FURTHER DEVELOPMENT OF LOCAL MQ-DQ METHOD AND ITS APPLICATION IN CFD

SHAN YONGYUAN

(B. Eng., Xi'an Jiaotong University, China)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2010

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor C. Shu, for his invaluable guidance, suggestions and patience throughout this study. His support and encouragement have contributed much towards the formation and completion of this dissertation.

I would also like to express my gratitude to all the staff members in the Fluid Mechanics Laboratory for their constant help and excellent service.

My gratitude also extends to my wife and my parents, whose support, patience and encouragement made it possible for me to complete this contribution.

Finally, I wish to express my appreciation to National University of Singapore for providing me with research scholarship, which makes this study possible.

Table of Contents

Summary

In the past two decades, a group of mesh-free methods were developed based on radial basis functions (RBFs). Local multiquadric-differential quadrature (MQ-DQ) method is a newly developed method which falls into this group. Compared with other RBF methods, the local MQ-DQ method mainly has two advantages. First, it is a local method, which makes it feasible to solve large scale problems. Second, it is based on derivative approximation instead of function approximation. Thus it can be well applied to both linear and nonlinear problems. The effectiveness of this method has been proven by its applications to various kinds of fluid flow problems. However, the research on the local MQ-DQ method is still in the preliminary stage. More work is required to further reveal its basic properties and improve its performance in solving fluid flow problems.

In this thesis, we firstly derived the formulas for the finite difference (FD) schemes based on the MQ function approximation instead of the low order polynomial approximation and named them as MQ-FD methods, which can be considered as special cases of the local MQ-DQ method. The effect of the shape parameter c in MQ on the formulas of the MQ-FD methods is analyzed. One interesting observation is that when c goes to infinity, the MQ-FD formulas of derivative approximation are the same as those given by the conventional FD

schemes. Another observation is that as compared with the conventional FD schemes, the MQ-FD methods may solve periodic boundary value problems more accurately. However, for general boundary value problems, the accuracy may not be as high as that using the conventional FD schemes.

Secondly, this thesis focused on improving the flexibility and efficiency of the local MQ-DQ method. An efficient local stencil adaptive algorithm was developed and combined with the local MQ-DQ method. The combined method bears the properties of both local MQ-DQ method for mesh-free numerical discretization and local stencil adaptive algorithm for high computational efficiency. Moreover, a hybrid technique which combines this mesh-free method with conventional FD scheme was adopted to further improve its efficiency. In this technique, the local MQ-DQ method is applied for the spatial discretization in the region around the curved boundary while conventional FD scheme is applied in the rest of the flow domain taking advantage of its high computational efficiency.

Finally, the local MQ-DQ method was extended to simulate fluid flow problems with curved boundary in three-dimensional (3D) space. An error estimate was provided for the 3D local MQ-DQ method to study the influence of shape parameter and the number of supporting points on its numerical accuracy. It was observed that the convergence rate can be improved by increasing the number of supporting points. The problem of flow past a sphere was simulated by the 3D

local MQ-DQ method to demonstrate its capability and flexibility in solving 3D fluid flow problems with curved boundary. The obtained numerical results showed that it is a promising scheme for solving 3D fluid flow problems with curved boundary.

List of Tables

List of Figures

Nomenclature

c	Shape parameter in radial basis function
C_p	Specific heat at constant pressure
g	Gravitational acceleration
k	Thermal conductivity
L	Reference length
p	Pressure
Pr	Prandtl number, $\Pr = \mu C_p / k$
Ra	Rayleigh number, $Ra = C_p \rho_0 g \beta L^3 \Delta T / k\nu$
Re	Reynolds number, $\mathrm{Re} = UL/\nu$
T	Temperature
t	Time
U	Reference velocity
u	Velocity component along x-direction
v	Velocity component along y-direction
w	Velocity component along z-direction
x	Cartesian coordinate
y	Cartesian coordinate
z	Cartesian coordinate
β	Thermal expansion coefficient

μ	Dynamic viscosity
ν	Kinematic viscosity
ρ_0	Reference density
ψ	Stream function
ω	Vorticity

Chapter 1 Introduction

1.1 Background

In computational fluid dynamics (CFD), the most popular numerical approaches used are finite difference (FD), finite volume (FV) and finite element (FE) methods. Many other methods are developed based on these three methods, which are thus regarded as traditional numerical methods. Despite the popularity of traditional methods, a number of new numerical schemes, such as mesh-free methods, have been developed in the past few decades. In the following, the traditional FD, FV, FE methods and the recently-developed mesh-free methods are briefly described.

1.1.1 Traditional numerical methods

FD method may be the oldest method for numerical solution of partial differential equations (PDEs). It could also be the easiest method for numerical computation. The fundamental idea of FD method is to approximate/interpolate the unknown functions by a local Taylor series expansion or polynomial fitting at grid points in the adopted mesh system. In practical implementations, the FD method essentially approximates the derivatives in the governing equations by a linear combination of values of dependent variables at a finite number of grid points along one line. Therefore, the most suitable computational domain for the FD method is the regular rectangular type, where it is accurate, efficient and simple to implement.

However, it does not adapt well to problems with complex geometry without appropriate coordinate transformation.

As compared with the FD method, FV method can accommodate any type of grid, so it is suitable for complex geometries. The FV method uses the integral form of the conservation equations as its starting point. The solution domain is subdivided into a finite number of contiguous control volumes, and the conservation equations are applied to each control volume. At the centroid of each control volume lies a computational node at which the variable values are to be calculated. Interpolation is used to express variable values at the control volume surface in terms of the nodal values. Surface and volume integrals are approximated using suitable quadrature formulae. As a result, one obtains an algebraic equation for each control volume, in which a number of neighboring nodal values appear. Since all terms that need be approximated in the FV method have physical meaning, it is very popular with engineers. As compared with the FD method, the disadvantage of the FV method appears in the three-dimensional applications, in which it is difficult for the FV method to achieve the accuracy of order higher than the second.

The FE method is similar to the FV method in many ways, except that the FE method solves the weak form of the partial differential equations. The solution domain is divided into a set of finite elements, which are generally unstructured to

fit the complex geometry. After its initial development from an engineering background, FE method has been formulated by mathematicians into a very elegant and strict framework, in which precise mathematical conditions for the existence of solution and convergence criteria and error bounds were well established. To fully understand the aspects of finite element discretization, appropriate mathematical background is needed for the end-users, such as functional analysis. The greater complexity of the FE method makes it more costly than the FD method in terms of computational cost.

1.1.2 Mesh-free methods

In recent decades, many new numerical schemes have been proposed to avoid the weakness of traditional numerical methods. Among them, a group of so-called mesh-free methods have especially attracted the attention of researchers. Mesh-free methods are a group of methods which can construct functional approximation or interpolation entirely from the information at a set of scattered nodes, among which there is no pre-specified connectivity or relationship. This means that they only require node generation instead of mesh generation, thus the computational costs associated with mesh generation are highly reduced, especially for problems with complex geometry. Another key advantage of mesh-free methods is the computational ease of adding and subtracting nodes from the pre-existing nodes. This property is particularly important for solving flow problems with large deformation or moving boundaries.

A number of mesh-free methods have been proposed up to now. Among them, a group of mesh-free methods, which are based on the so-called radial basis functions (RBFs), have received increasing attention by researchers in the past two decades. RBFs are a primary tool for interpolating multi-dimensional scattered data. Due to their "mesh-free" property, RBFs were adopted to deal with PDEs in recent years. In the following section, we will give a literature review on the development of RBFs methods, especially for multiquadric (MQ) RBFs.

1.2 Literature review on function and derivative approximation by RBFs

1.2.1 Radial basis functions (RBFs)

A radial basis function is a continuous spline which depends on the separation distances of a subset of scattered points. There are many RBFs available. The most commonly used RBFs are

Multiquadrics (MQ): $\varphi(r) = \sqrt{r^2 + c^2}$ (1.1a)

Thin-plate splines (TPS): $\varphi(r) = r^2 \log(r)$ (1.1b)

Gaussians: $\varphi(r) = e^{-cr^2}$ (1.1c)

Inverse multiquadrics (MQ): $\varphi(r) = \dfrac{1}{\sqrt{r^2 + c^2}}$ (1.1d)

where $r = \left\| \mathbf{x} - \mathbf{x}_j \right\|_2$ and shape parameter c is a positive constant.

Among the above popular RBFs, the Gaussian and the inverse MQ are positive

definite functions, while the MQ and the TPS are conditionally positive definite functions. From the test result of interpolation by Franke (1982), it was found that MQ obtained the best solution in accuracy, and TPS ranked the second. However, though TPS RBFs have been considered as one of the optimal functions for multivariate data interpolation, they do only converge linearly (Powell, 1994). Compared to TPS RBFs, the MQ functions can converge exponentially and always produce a minimal semi-norm error (Madych, 1990).

1.2.2 Interpolation by MQ RBFs

The multiquadric (MQ) interpolation method was first developed by Hardy (1971) to produce topographic maps based upon elevations at arbitrarily located points in a plane. Hardy's basic scheme is very simple and easy to implement. It is assumed that any function, *f*, may be written as an expansion of *N* continuously differentiable radial basis functions, *g*:

$$f(\mathbf{x}) = \sum_{j=1}^{N} a_j g_j(\mathbf{x} - \mathbf{x}_j) \tag{1.2}$$

where

$$g_j(\mathbf{x} - \mathbf{x}_j) = \sqrt{d_j^2(\mathbf{x} - \mathbf{x}_j) + c^2} \tag{1.3}$$

c^2 is a non-zero input parameter, and

$$d_j^2(\mathbf{x} - \mathbf{x}_j) = (x - x_j)^2 + (y - y_j)^2 + \cdots \tag{1.4}$$

The basis functions for MQ method consist of the sum of the square of the Euclidean distance between a data point and the evaluation point, and a non-negative parameter c^2. The value of c^2 controls the shape of the basis

function. Large c^2 values give rise to flat sheet-like basis functions, intermediate c^2 values give rise to bowl-like basis functions and small c^2 values give rise to narrow cone-like basis functions.

The coefficients $\{a_j\}$ are found by solving a set of linear equations in terms of the basis functions.

$$\sum_{j=1}^{N} a_j g_j(\mathbf{x}_i - \mathbf{x}_j) = F(\mathbf{x}_i), \qquad i = 1, 2, ..., N \tag{1.5}$$

and $f(\mathbf{x}_i) = F(\mathbf{x}_i)$ which are given.

The MQ method was applied successfully in various early applications, but it was almost unknown by mathematicians. This situation was changed when Franke (1982) published a review paper which evaluated 29 different algorithms for the scattered data interpolation problem on a variety of known data surfaces. Franke graded various scattered data interpolation schemes according to the following criteria: accuracy, visual aspect, sensitivity to parameters, execution time, storage requirements and ease of implementation. Franke stated that of all the methods tested, Hardy's MQ method (1971) gave the most accurate results. The second best interpolation method was the thin-plate spline of Duchon (1976). Consequently, the MQ method began to be popular among researchers.

Stead (1984), like Franke (1982), examined various methods for estimating partial derivatives on scattered data. She concluded that for surfaces with large curvature,

MQ is excellent for obtaining very accurate derivative estimates, but that MQ behaves poorly on relatively flat surfaces. She therefore recommended a combination of techniques in which MQ would be applied to steep surfaces and a quadratic fit would be used for relatively flat surfaces without using transformations such as stretching functions which cause a change in geometry.

Madych and Nelson (1990) considered a general class of interpolants which includes a polynomial of degree less than some fixed integer *m* given by

$$f(\mathbf{x}) = \sum_{j=1}^{N} a_j g(\mathbf{x} - \mathbf{x}_j) + \sum_{|\alpha|<\mathbf{m}} k_\alpha \mathbf{x}^\alpha \,, \tag{1.6}$$

where a_j and k_α must satisfy

$$\sum_{j=1}^{N} a_j g(\mathbf{x}_i - \mathbf{x}_j) + \sum_{|\alpha|<\mathbf{m}} k_\alpha \mathbf{x}_i^\alpha = f(\mathbf{x}_i) = F_i \,, \quad i = \{1, 2, ..., N\} \,, \tag{1.7a}$$

$$\sum_{j=1}^{N} a_j \mathbf{x}_j^\alpha = 0 \,, \quad |\alpha| < m \,. \tag{1.7b}$$

This type of MQ interpolant can achieve polynomial precision.

The mathematical analysis of MQ is very difficult, and it is not known why MQ performs so well. Initially, many mathematicians found the MQ method to be enigmatic, counter-intuitive, and difficult to analyze. Later, several theoretical advances have been established. Micchelli (1986) proved that the linear system obtained from the interpolation conditions is always solvable for distinct data. He has shown that MQ coefficient matrix of rank *N* has one positive real eigenvalue and (*N*-1) negative real eigenvalues. Furthermore, he has shown that Duchon's

thin-plate spline is a positive definite interpolant and Hardy's MQ interpolant is conditionally positive definite. The MQ interpolant can be positive definite by appending linear polynomials. Madych and Nelson (1990) have shown that for all functions in the space of conditionally positive-definite functions, a semi-norm exists which is minimized by all such functions.

Tarwater (1985) investigated MQ to determine the effect of varying the parameter c^2 on the goodness of fit. Various experiments were tried to obtain increasingly more accurate interpolants. It was found that the root mean square (r.m.s) error was a function of the magnitude of c^2; for increasing c^2, the errors gradually dropped to a minimum at the so-called optimum c^2, and then grew rapidly thereafter. She showed that for the several problems examined, the r.m.s errors of the MQ interpolation compared favorably or even outperformed the monotonic cubic spline in terms of accuracy. By adjusting the parameter c^2, she found that the accuracy could be considerably improved. The most difficult function to interpolate was the steep "cliff" function. She suggested that the shape of the surface to be fitted is a factor in optimizing the accuracy. Lancaster and Salkauskas (1986) also noted that noisy surfaces corresponding to the ill-conditioning of the MQ coefficient matrix can occur if *c* becomes large.

In the effort to improve the performance of MQ, Kansa (1990) extended the original Hardy scheme in three areas. First, he permitted the basis function shape

parameter c^2 to vary monotonically. This gives a set of basis functions whose shapes vary from flat sheets to rounded cones in R^n. As a result, he simultaneously improved the accuracy and reduced the condition number of the coefficient matrix. Second, he used domain decomposition and blending to change a global surface fitting problem into overlapping quasi-local problems. Such decomposition has the additional benefit of rendering the quasi-local MQ coefficient matrix better conditioned with smaller rank. Third, he found that the "track" data problem (data which is closely spaced in one direction and widely spaced in the orthogonal directions) can be treated accurately by transformations which "randomize" the transformed independent variables.

1.2.3 Kansa's MQ collocation method for solving PDE

Due to excellent performance of MQ in scattered data interpolation, Kansa (1990) got the idea of extending it to solve partial differential equations. He advocated the use of MQ as a tool for computational fluid-dynamics due to the following reasons. MQ is a very high order continuously differentiable scheme which performs well on scattered and gridded data. Fewer points are required in MQ than in low order finite difference or element schemes. Tensor product meshes are not required in higher dimensions, thus simplifying problems with irregular physical boundaries. MQ does not have the connectivity restrictions associated with local finite difference or element schemes. Moving node and Lagrangian schemes based on such local methods with a specific connectivity may give

problems with tangled zones or negative areas and volumes. Points may be added or deleted simply using MQ since connectivity is not a problem.

Kansa (1990) adopted MQ as a spatial approximation scheme for parabolic, hyperbolic and elliptic partial differential equations. He found that MQ is not only exceptionally accurate, but is more efficient than finite difference schemes which require many more operations to achieve the same degree of accuracy. In Kansa's method, partial differential equations are solved by collocation with MQ RBFs. It rewrites the problem as a generalized interpolation problem, and the solution is obtained by solving a (possibly large) linear system.

To show the procedure of Kansa's collocation method clearly, we will consider the solution of a PDE given below

$$Lu = f \quad \text{in} \quad \Omega \subset \Re^d \tag{1.8a}$$

$$u = g \quad \text{on} \quad \partial\Omega \tag{1.8b}$$

where the operator L is the linear partial differential operator.

The solution u of the PDE approximated by MQ can be written as a linear combination of N radial basis functions, viz.

$$u(\mathbf{x}) \approx \sum_{j=1}^{N} \lambda_j \varphi(\mathbf{x}, \mathbf{x}_j) \quad \text{for} \quad \mathbf{x} \in \Omega \subset \Re^d \tag{1.9}$$

Substituting equation (1.9) into equation (1.8) gives

$$Lu(\mathbf{x}_i) \approx \sum_{j=1}^{N} \lambda_j L\varphi(\mathbf{x}_i, \mathbf{x}_j) = f(\mathbf{x}_i), \qquad i = 1, \ldots, N - M \tag{1.10a}$$

$$u(\mathbf{x}_i) \approx \sum_{j=1}^{N} \lambda_j \varphi(\mathbf{x}_i, \mathbf{x}_j) = g(\mathbf{x}_i), \qquad i = N - M + 1, \ldots, N \tag{1.10b}$$

where M is the number of boundary points and N is the total number of nodes.

Expressing equation (1.10) in matrix form, we have the following expression:

$$\underbrace{\begin{bmatrix} L\varphi(x_1, x_1) & \cdots & L\varphi(x_1, x_N) \\ \vdots & \vdots & \vdots \\ L\varphi(x_{N-M}, x_1) & \cdots & L\varphi(x_{N-M}, x_N) \\ \varphi(x_{N-M+1}, x_1) & \cdots & \varphi(x_{N-M+1}, x_N) \\ \vdots & \vdots & \vdots \\ \varphi(x_N, x_1) & \cdots & \varphi(x_N, x_N) \end{bmatrix}}_{[W]} \underbrace{\begin{Bmatrix} \lambda_1 \\ \vdots \\ \lambda_N \end{Bmatrix}}_{c} = \underbrace{\begin{Bmatrix} f(x_1) \\ \vdots \\ f(x_{N-M}) \\ g(x_{N-M+1}) \\ \vdots \\ g(x_N) \end{Bmatrix}}_{b} \tag{1.11}$$

We can rewrite the equation (1.11) in the following matrix form

$$[W]c = b \tag{1.12}$$

which results in the solution as

$$c = [W]^{-1} b \tag{1.13}$$

Once we get the solution for c, the solution for PDE equation (1.8) is expressed as equation (1.9).

From the feasibility point of view, we need to make sure that the coefficient matrix W is invertible. Hon and Schaback (2001) showed that there cannot be a general proof of non-singularity of matrices arising from asymmetric collocation with radial basis functions. However, they also pointed out that the pure existence of singular cases is no serious objection to a valuable numerical technique. There are reliable techniques to detect near-singularity of matrices, and if these

techniques are incorporated into running code, applications are safeguarded.

After Kansa's work (1990), more and more researchers showed their interest in the implementation of RBFs in PDE solvers and made considerable achievements. Since the resultant coefficient matrix of Kansa's method is not symmetric, Fasshauer (1997) and Jumarhon et al. (2000) presented an alternative approach based on the Hermite interpolation property of the radial basis functions, which states that the RBFs not only are able to interpolate a given function, but also its derivatives. Another benefit which can also be enjoyed from Hermite RBF scheme is the symmetric coefficient matrix, which guarantees the related linear equations being solvable. Franke and Schaback (1998) proved the convergence of the symmetric collocation method. They provided their first theoretical foundation for solving PDEs by RBF collocation method and considered it a special case of the general Hermite-Birkhoff interpolation problem. Wu (1998) gave the convergence proofs for the use of Hermite-Birkhoff RBF in solving PDEs.

It is known that RBF approximation tends to degrade in the boundary neighboring regions. Chen (2002) pointed out that one common issue in Kansa's method and symmetric Hermite method is that the numerical solutions at nodes adjacent to the boundary deteriorate (by one to two orders) as compared with those in the central region. Taking this into consideration, he proposed a new RBF collocation approach based on Kansa's method to improve the solution accuracy near the

boundary.

Cheng et al. (2003) presented a so-called H-c multiquadric collocation method, which showed exponential convergence by numerical experiments.

Hon et al. further extended the use of Kansa's method on the numerical solutions of various ordinary and partial differential equations. They applied this method to solve general initial value problems (Hon et al., 1997), nonlinear Burger's equation with shock wave (Hon et al., 1998), surface wind field computation from scattered data (Hickernell and Hon, 1998), complicated biphasic and triphasic models of mixtures (Hon et al., 1997), shallow water equation for tide and currents simulation under irregular boundary (Hon et al., 1999) and free boundary problems like American option pricing (Hon et al., 1999). The computations showed the definite advantages in using this truly mesh-free method for solving various initial and boundary value problems.

1.2.4 Drawbacks of MQ collocation method

Kansa's collocation method usually encounters two difficulties in practical applications. One difficulty arises from the fact that radial basis functions are globally used in the collocation technique to solve PDEs. In other words, the support of every node covers the whole domain. Consequently, the system of algebraic equations generated by the collocation method usually has very large

condition number, and becomes increasingly ill-conditioned as the number of nodes increases. As observed by Dubal et al (1993) and Driscoll and Fornberg (2002), the coefficient matrix arising from using about 2000 knots is extremely ill-conditioned. Therefore, when complex problems are confronted and a large number of collocation nodes are required to catch the physical details, the problem of ill-conditioning is almost unavoidable. It can be seen that full exploitation of the advantage of the RBFs-based methods has been hampered by the progressively more ill-conditioned coefficient matrix as the rank increases. As a consequence, the pure global collocation method is of little practical use in engineering. To remove this difficulty, Kansa and his co-workers have made a lot of attempts. Kansa and Hon (2000) proposed a domain decomposition method, which can reduce many orders of magnitude of the condition number of resultant linear equations. Ling and Kansa (2005) proposed a least-squares preconditioning scheme to transform a badly conditioned linear system into one that is very well conditioned.

On the other hand, Kansa's scheme is actually based on the function approximation. In other words, it directly substitutes the expression of function approximation by RBFs into a PDE, and then changes the dependent variables into the coefficients of function approximation. The process is very complicated, especially for non-linear problems. This may be another reason for which the method has not been extensively applied to solve practical problems.

1.2.5 Local MQ-DQ method

To overcome the drawbacks of Kansa's collocation method, Shu et al. (2003) proposed a novel meshless method named local RBF-DQ method. This method can be regarded as a combination of the conventional differential quadrature (DQ) method with the RBFs by means of taking the RBFs as the trial functions in the DQ scheme. As a result, it combines the mesh-free property of RBFs approximation with the derivative approximation of DQ method. Since MQ RBFs are mainly used in their study, their method is also known as local MQ-DQ method.

1.2.5.1 Differential quadrature (DQ) method

The DQ method is a numerical discretization technique for approximation of derivatives. It was first proposed by Bellman et al. (1971, 1972). The essence of the DQ method is that the partial derivative of an unknown function with respect to an independent variable is approximated by a weighted linear sum of function values at all discrete points within its support. Suppose that a function $f(x)$ is sufficiently smooth. Then its mth order derivative with respect to x at a point x_i can be approximated by DQ as

$$\left.\frac{\partial^m f}{\partial x^m}\right|_{x_i} = \sum_{j=1}^{N} w_{ij}^{(m)} f(x_j), \quad i = 1, 2, ..., N, \tag{1.14}$$

where x_j are the discrete points in the domain, $f(x_j)$ and $w_{ij}^{(m)}$ are the function values at these points and the related weighting coefficients. Obviously,

the key procedure in the DQ method is the determination of the weighting coefficients $w_{ij}^{(m)}$. It has been shown by Shu (2000) that the weighting coefficients can be easily computed under the analysis of a linear vector space and the analysis of a function approximation. When the function is approximated by a high order polynomial, Shu and Richards (1992) derived a simple algebraic formulation and a recurrence relationship to compute the weighting coefficients of any order derivative, and the method is termed as PDQ. When the function is approximated by a Fourier series expansion, Shu and Chew (1997) further derived simple algebraic formulations to compute the weighting coefficients of the first and second order derivatives, and the method is called FDQ. It should be indicated that both the PDQ and FDQ methods are applied along a mesh line, that is, the functional values are taken at points on a mesh line. This is because in these methods, the functional approximation is actually one-dimensional.

1.2.5.2 Local MQ-DQ method

Different from the above conventional DQ method, MQ RBFs were taken by Shu et al. (2003) as the trial functions in the DQ scheme to determine the weighting coefficients in equation (1.14). Due to the mesh-free property of MQ approximation, the MQ RBF based DQ method is also mesh-free. This is the major difference between the conventional DQ and MQ-DQ methods.

Compared with Kansa's collocation method, the local MQ-DQ method mainly has

two advantages. The first one is that it is a local method. As compared with domain decomposition method, this method not only can be employed with a large number of nodes, but also requires no extra effort on the division of the computational domain. The other one is that, due to the usage of DQ technique, the coefficients computed by the DQ technique can be equally well applied to the linear and nonlinear problems.

It is known that the performance of the local MQ-DQ method depends on three factors, i.e. local density of knots, free shape parameter and number of supporting knots. Due to the lack of theoretical analysis, it is difficult for the end-users to know how these factors will affect its performance and how to choose them. Ding et al. (2005) carried out an error estimate of the local MQ-DQ method through numerical experiments to investigate the contributions of these three factors to the approximation error and their correlations. Their work can serve as a useful guidance for the implementation of the local MQ-DQ method.

Up to now, the capability and flexibility of the local MQ-DQ method have been demonstrated by applying it to solve different kinds of fluid flow problems. Firstly, it was applied to solve several 2D incompressible viscous fluid flow problems, such as natural convection (Shu et al., 2003 and Ding et al., 2005), and driven cavity flow and flow past a circular cylinder (Shu et al., 2005). Shu et al. (2005) also developed an upwind local RBF-DQ method for simulation of inviscid

compressible flows. Furthermore, the local MQ-DQ method was extended to solve incompressible viscous fluid flow problems in 3D space (Ding et al, 2006). All the above applications showed that the local MQ-DQ method is an effective and efficient numerical method for fluid flow problems.

Recently, Tolstykh and Shirobokov (2003) also proposed a RBFs-based derivative approximation scheme, in which an approximate formula for the derivative discretization is constructed based on the local RBF-interpolants. Due to the similarity between the local supports and the stencils in finite difference methods, the approach is regarded as using radial basis functions in a “finite difference mode”. This approach is very similar to the local RBF-DQ method. The main difference between the two methods is that the local RBF-DQ formulations are derived from the concept of differential quadrature while the approach of Tolstykh and Shirobokov (2003) is constructed from the idea of finite difference schemes. Similar work was also done by Wright and Fornberg (2006). In their work, they constructed scattered node compact finite difference-type formulas through RBFs.

1.3 Objective of this thesis

1.3.1 Motivations

Although the effectiveness and flexibility of the local MQ-DQ method has been proven through its application to simulate many kinds of fluid flow problems, the study on the development of this novel method is still in the preliminary stage.

Firstly, the relationship between this novel mesh-free method and the conventional numerical methods is not known to researchers. It will be very helpful for our better understanding of this mesh-free method if its relationship to the conventional numerical methods is revealed. Secondly, as compared with the conventional numerical methods such as FD, FE and FV method, the local MQ-DQ method is far from efficient during computation. From real application point of view, it is important to improve its efficiency. Thirdly, most applications of the local MQ-DQ method are focused on two-dimensional fluid flow problems. Study of this method for 3D cases is very rare and simple. It is interesting to know how well this method can perform for 3D cases.

1.3.2 Objectives

The objective of this thesis is to further study the performance of the local MQ-DQ method and make it more efficient in real applications. The principal goals of the research are as follows:

a. To study the relationship between the local MQ-DQ method and the traditional finite difference method.
b. To develop an adaptive local MQ-DQ method which is efficient and flexible for 2D fluid flow problems with curved boundary.
c. To improve the efficiency of the local MQ-DQ method through combining it with traditional finite difference method.
d. To study the performance of local MQ-DQ method in 3D space and apply it to

solve 3D fluid flow problems with curved boundary.

1.4 Organization of this thesis

The outline of this thesis is as follows:

Chapter 2 gives out the governing equations for incompressible viscous flows in both primitive variable formulation and stream function-vorticity formulation. In addition, spatial and temporal discretization methods are presented for the solution of these governing equations.

In **Chapter 3**, we derive the formulation of the local MQ-DQ method with the stencil of conventional FD scheme and name it as MQ-FD method. The relationship between this meshless method and conventional FD scheme is created. Performance of the MQ-FD method is studied through numerical experiments.

In **Chapter 4**, we propose a local stencil adaptive technique for the local MQ-DQ method. This adaptive technique improves the efficiency of the local MQ-DQ method and helps to better capture the properties of the numerical solutions. Two numerical experiments are carried out to validate the effectiveness and efficiency of this adaptive meshless method.

In **Chapter 5**, a hybrid technique is proposed to improve the efficiency of the local MQ-DQ method. This hybrid technique combines the local MQ-DQ method

with the central FD scheme and is applied to simulate flows past a circular cylinder. In addition, an empirical formula is proposed to determine its shape parameter.

In **Chapter 6**, the local MQ-DQ method is extended to simulate fluid flow problems with curved boundary in 3D space. An error estimate for the 3D local MQ-DQ method is carried out to provide a useful guidance for its implementation. Based on the observation, this meshless method is combined with the conventional FD method for the improvement of its efficiency.

Chapter 7 gives a summary of the most important conclusions. The recommendations for future research work are also proposed.

Chapter 2 Governing Equations and Solution Methods

Fluid flows are governed by a set of partial differential equations (PDEs), which are derived from the conservation laws of mass, momentum and energy. For Newtonian fluids, the governing equations are known as Navier-Stokes (N-S) equations. Since the N-S equations are non-linear, coupled and difficult to solve, some simplifications have been introduced to reduce its expression. With different simplifications, fluid flows can be classified into various groups. With the assumption of constant fluid density, flows are considered to be incompressible and the governing equations will be simplified to be incompressible N-S equations. If viscous effects are neglected, flows are considered to be inviscid and the governing equations are simplified to be Euler equations. With other simplifications, many other types of reduced governing equations can be derived, such as potential flow and creeping flow. In this study, we mainly focus on incompressible viscous flows. In the following, the governing equations of the incompressible viscous flows will be given out, followed by the solution methods for these governing equations.

2.1 Governing equations for incompressible viscous fluid flows

For incompressible fluid flow problems, there are two commonly used formulations of the governing equations: primitive variable (velocity and pressure) formulation and stream function-vorticity formulation. Both formulations are used

widely and have their own advantages. In the following, only the two-dimensional governing equations will be presented for the convenience of showing the stream function-vorticity formulation. The energy equation is omitted for general non-thermal fluid flow problems. Governing equations in 3-D space and governing equations for thermal flow problems will be presented when necessary in the following chapters.

2.1.1 Primitive variable formulation

The non-dimensional governing equations in the primitive variable form are usually written as

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{\partial p}{\partial x}+\frac{1}{\mathrm{Re}}(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}) \tag{2.1}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{\partial p}{\partial y}+\frac{1}{\mathrm{Re}}(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}) \tag{2.2}$$

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \tag{2.3}$$

where u and v are the velocity components along the x-direction and y-direction, respectively. p is the pressure. Re is the dimensionless Reynolds number and is defined as $\mathrm{Re}=\frac{UL}{\upsilon}$, where U is characteristic velocity, L is reference length and υ is known as the kinematic viscosity.

The extension of above governing equations to the three-dimensional space is straightforward. The primitive variable formulation is the most popular one, especially in three-dimensional flow simulations.

Solution of above equations encounters difficulties like the lack of an independent equation for the pressure and non-existence of a dominant variable in the continuity equation. One way to circumvent these difficulties is to decouple the pressure computation from the momentum equations and then construct a pressure field so as to enforce the satisfaction of the continuity equation. Details of this solution method will be discussed in the following chapters. Another way is to take away the pressure term from the governing equations and to construct a so-called stream function-vorticity formulation. Details are shown in the following section.

2.1.2 Stream function-vorticity formulation

By taking the derivative of equation (2.1) with respect to *y* and taking the derivative of equation (2.2) with respective to *x*, we can get the following expressions,

$$\frac{\partial}{\partial y}[\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}]=\frac{\partial}{\partial y}[-\frac{\partial p}{\partial x}+\frac{1}{\mathrm{Re}}(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2})] \tag{2.4}$$

$$\frac{\partial}{\partial x}[\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}]=\frac{\partial}{\partial x}[-\frac{\partial p}{\partial y}+\frac{1}{\mathrm{Re}}(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2})] \tag{2.5}$$

To take away the pressure term from the governing equations, we subtract equation (2.4) from equation (2.5) and get the following expression

$$\frac{\partial}{\partial t}(\frac{\partial u}{\partial y}-\frac{\partial v}{\partial x})+u\frac{\partial}{\partial x}(\frac{\partial u}{\partial y}-\frac{\partial v}{\partial x})+v\frac{\partial}{\partial y}(\frac{\partial u}{\partial y}-\frac{\partial v}{\partial x})$$
$$=\frac{1}{\mathrm{Re}}[\frac{\partial^2}{\partial x^2}(\frac{\partial u}{\partial y}-\frac{\partial v}{\partial x})+\frac{\partial^2}{\partial y^2}(\frac{\partial u}{\partial y}-\frac{\partial v}{\partial x})] \quad (2.6)$$

Defining $\omega=\frac{\partial u}{\partial y}-\frac{\partial v}{\partial x}$ and substituting it into equation (2.6), we can get

$$\frac{\partial \omega}{\partial t}+u\frac{\partial \omega}{\partial x}+v\frac{\partial \omega}{\partial y}=\frac{1}{\mathrm{Re}}(\frac{\partial^2 \omega}{\partial x^2}+\frac{\partial^2 \omega}{\partial y^2}) \quad (2.7)$$

ω is called vorticity of the flow. It is a very interesting characteristic of the flow.

The vorticity can be combined with the stream function to yield a concise description of the governing equations. The stream function is related to the velocity by

$$u=\frac{\partial \psi}{\partial y} \quad (2.8)$$

$$v=-\frac{\partial \psi}{\partial x} \quad (2.9)$$

With these expressions, the continuity equation (2.3) is automatically satisfied. Substituting equations (2.8) and (2.9) into the vorticity formulation, we can lead to the relation between vorticity and stream function as

$$\frac{\partial^2 \psi}{\partial x^2}+\frac{\partial^2 \psi}{\partial y^2}=\omega \quad (2.10)$$

Equations (2.7)-(2.10) provide a complete description of a two-dimensional incompressible flow and are named as stream function-vorticity formulation. There is no pressure term in these equations. Its solution procedure is simpler as compared with that of primitive variable formulation, thus is widely used in 2-D

fluid flow problems. On the other hand, the extension of this approach to three dimensional flows requires the introduction of additional stream functions which makes the formulations even more complicated than its primitive variable counterpart. In addition, the boundary conditions for vorticity are not straightforward.

2.2 Solution methods

Because of the complexity of the governing equations, it is impractical, if not impossible, to obtain their solutions analytically. Instead, these governing equations can be solved numerically. To numerically solve these governing equations, we first need to discretize the partial differential equations into a set of algebraic equations. The resulting algebraic equations can be solved either by direct methods such as Gauss Elimination, LU decomposition, or by iterative methods such as Gaussian-Seidal, Successive-Over-Relaxation (SOR).

In the governing equations, there are both derivatives with respect to space (x, y) and time (t). The discretization of derivatives with respect to space is called spatial discretization and discretization of derivatives with respect to time is called temporal discretization. In the following, the numerical methods used for spatial and temporal discretization in this work will be described in detail.

2.2.1 Spatial discretization method: local MQ-DQ method

As its name implies, the local MQ-DQ method is based on the MQ RBFs and DQ technique. The concept of DQ was first proposed by Bellman et al. (1971, 1972) to approximate the derivative of a smooth function. From the viewpoint of derivative approximation, the essence of the DQ method is that the partial derivative of any dependent variable can be approximated by a weighted linear sum of functional values at all discrete points within its support. In other words, the DQ approximation of the *m*th order derivative of a function $f(x)$ at x_i can be expressed as

$$\left.\frac{\partial^m f}{\partial x^m}\right|_{x=x_i} = \sum_{j=1}^{N_i} w_{i,j}^{(m)} f(x_j), \tag{2.11}$$

where x_j denotes the coordinates of discrete points in the support of reference node *i*, $f(x_j)$ and $w_{i,j}^{(m)}$ are the function values at these points and the related weighting coefficients. It should be noted that the subscript *i* is a global nodal index which represents the nodes in the domain, while the subscript *j* is a local index which represents the supporting points for the reference node *i*. Since the reference node is also one of its supporting points, we define $(\cdot)_{i,1} = (\cdot)_i$ for simplicity. N_i is the number of supporting points within the support for the reference node *i*. To determine the weighting coefficients $w_{i,j}^{(m)}$ in equation (2.11), a set of trial functions are required. In the present local MQ-DQ method, the MQ RBFs are selected for its mesh-free property and super convergence.

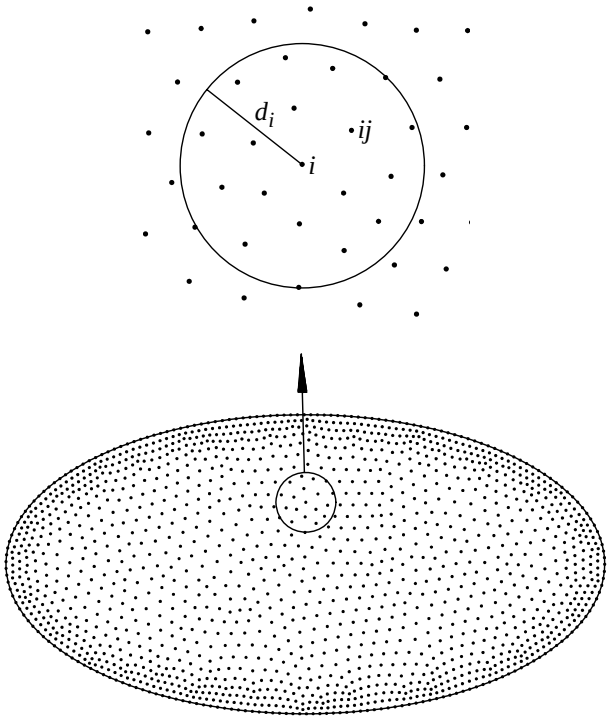


Figure 2.1 Supporting points around a reference point

Consider a sub-domain Ω_{x_i} as shown in Figure 2.1, which forms the neighborhood or support of a reference point x_i and denotes as the domain of function approximation at x_i by the trial functions. To approximate the function u in Ω_{x_i} over a number of randomly-located supporting points, the RBF approximant $u^h(x)$, can be defined by a linear sum of MQ RBFs as

$$u^h(x) = \sum_{j=1}^{N_i} \lambda_j \varphi(\|x, x_j\|), \tag{2.12}$$

where $\varphi(\|x, x_j\|) = \sqrt{r_j^2 + c^2}$ denotes the MQ RBF at x_j, and x_j, $j = 1, 2, ..., N_i$, are the supporting points in Ω_{x_i}. r represents the Euclidean norm and c is a positive free shape parameter given by the end-user. N_i denotes the total number of supporting points in Ω_{x_i}. λ_j denotes the corresponding constant coefficients. Equation (2.12) can be considered to construct an N_i dimensional vector space V_{N_i}. Obviously, the set of MQ RBFs $\varphi(\|x, x_j\|)$, $j = 1, 2, ..., N_i$,

form the base functions (vectors). It can be seen that if all the base functions $\varphi(\|x, x_j\|)$ satisfy a linear relationship like equation (2.11), so does any function in V_{N_i}. Therefore, equation (2.11) can be well applied to discretize the derivatives of function $u(x)$. In the following, the first-order derivative is taken as an example to show the procedure for the determination of weighting coefficients.

Substituting the set of MQ RBFs (base vectors in V_{N_i}) into equation (2.11), the determination of corresponding coefficients for the first-order derivative is equivalent to solving the following linear equations:

$$\frac{\partial \varphi_k(x_i)}{\partial x} = \sum_{j=1}^{N_i} w_{i,j}^{(1x)} \varphi_k(x_j) \qquad k = 1, 2, ..., N_i . \tag{2.13}$$

For simplicity, the notation $\varphi_k(x)$ is adopted to replace $\varphi(\|x, x_k\|)$ in equation (2.13). The above equations can be further written in the matrix form,

$$\underbrace{\begin{bmatrix} \frac{\partial \varphi_1(x_i)}{\partial x} \\ \frac{\partial \varphi_2(x_i)}{\partial x} \\ \vdots \\ \frac{\partial \varphi_{N_i}(x_i)}{\partial x} \end{bmatrix}}_{\left\{\frac{\partial \varphi(x_i)}{\partial x}\right\}} = \underbrace{\begin{bmatrix} \varphi_1(x_1) & \varphi_1(x_2) & \cdots & \varphi_1(x_{N_i}) \\ \varphi_2(x_1) & \varphi_2(x_2) & \cdots & \varphi_2(x_{N_i}) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{N_i}(x_1) & \varphi_{N_i}(x_2) & \cdots & \varphi_{N_i}(x_{N_i}) \end{bmatrix}}_{[A]} \underbrace{\begin{bmatrix} w_{i,1}^{(1x)} \\ w_{i,2}^{(1x)} \\ \vdots \\ w_{i,N_i}^{(1x)} \end{bmatrix}}_{\{w\}} \tag{2.14}$$

Clearly, there exists a unique solution only if the matrix [A] is non-singular. The non-singularity of the matrix [A] depends on the properties of the RBFs used. According to the work of Micchelli (1986), matrix [A] is conditionally positive definite for MQ RBFs. This fact cannot guarantee the non-singularity of matrix [A]. Hon and Schaback (2001) showed that cases of singularity are quite rare, and

do not pose as serious objections to a valuable numerical technique. Therefore, the coefficient vector $\{w\}$ can be obtained by

$$\{w\} = [A]^{-1}\{\frac{\partial \varphi(x_i)}{\partial x}\}. \tag{2.15}$$

Now, we can use the coefficient vector $\{w\}$ to approximate the first-order derivative in the *x*-direction for any unknown function at node x_i. The calculation of weighting coefficients for other derivatives follows the same procedure.

From the procedure of DQ approximation for derivatives, it can be observed that the weighting coefficients are only dependent on the selected RBFs and the distribution of the supporting points in the support domain Ω_{x_i}. In numerical simulation, they are only computed once, and stored for all numerical discretization. Once the coefficients are computed, they will be stored and used to discretize the partial differential equation in a similar manner as in the traditional finite difference (FD) method. It should be noted that the computed coefficients can be consistently well applied to linear and non-linear problems. Therefore, it is very convenient to use local MQ-DQ method to solve complex non-linear problems such as Navier-Stokes equations in fluid mechanics. From the above, we can also see that the implementation of local MQ-DQ method is very simple and straightforward.

2.2.2 Temporal discretization method

Temporal discretization method is used to discretize the derivatives with respect to

time in the governing equations. There are various kinds of temporal discretization methods, such as first-order Euler method (forward and backward), second-order Crank-Nicolson method, second-order Adams-Bashforth method and fourth-order Runge-Kutta method. Among them, the fourth-order Runge-Kutta method and second-order Crank-Nicolson method will be used in this thesis, thus we will focus on these two methods.

To show the formula of the temporal discretization methods, the following ordinary differential equation (ODE) is taken as a model problem:

$$\frac{dy}{dt} = f(t, y), \qquad 0 \le t \le T \tag{2.16a}$$

$$y(0) = \alpha \tag{2.16b}$$

where *t* represents time and $y = y(t)$.

The fourth-order Runge-Kutta method is the most popular time discretization method. It is an explicit method, thus is easy to program and is conditionally stable. It is a good choice due to its high accuracy, good stability and ease to program. Starting with the initial point (t_0, y_0), the fourth-order Runge-Kutta method predicts the sequential solutions by the following formula:

$$y_{k+1} = y_k + \frac{h}{6}(f_1 + 2f_2 + 2f_3 + f_4) \tag{2.17}$$

where $f_1 = f(t_k, y_k)$

$$f_2 = f(t_k + \frac{h}{2}, y_k + \frac{h}{2} f_1)$$

$$f_3 = f(t_k + \frac{h}{2}, y_k + \frac{h}{2} f_2)$$

$$f_4 = f(t_k + h,\ y_k + hf_3)$$

h is the time step length.

The second-order Crank-Nicolson method is an implicit method. Thus, it is unconditionally stable. Starting with the initial point (t_0, y_0), it predicts the sequential solutions by the following formula:

$$y_{k+1} = y_k + \frac{h}{2}[f(t_k, y_k) + f(t_{k+1}, y_{k+1})] \qquad (2.18)$$

Chapter 3 Multiquadric Finite Difference (MQ-FD) Method and Its Application

The conventional finite difference (FD) schemes are based on the low order polynomial approximation in a local region. This chapter shows that when the polynomial approximation is replaced by the multiquadric (MQ) function approximation in the same region, new FD schemes, which are termed as MQ-FD methods in this work, can be developed. This chapter gives analytical formulas of the MQ-FD method and carries out a performance study for its derivative approximation and solution of Poisson equation and incompressible Navier-Stokes equations. In addition, the effect of the shape parameter c in MQ on the formulas of the MQ-FD method is analyzed. Derivative approximation in one-dimensional space and Poisson equation in two-dimensional space are taken as model problems to study the accuracy of the MQ-FD method. Furthermore, a lid-driven flow problem in a square cavity is simulated by the MQ-FD method. The obtained results indicate that this method may solve the engineering problem very accurately with a proper shape parameter c.

3.1 Motivation of this work

As shown in Chapter 2, before the implementation of the local MQ-DQ method to solve partial differential equations, we need to compute the weighting coefficients first. In numerical simulation, the weighting coefficients are only computed once,

and stored for all numerical discretization. Once the coefficients are computed, they will be stored and used to discretize the partial differential equations in a similar manner as in the conventional FD scheme. Thus, it can be seen that the implementation of the local MQ-DQ method is very simple and straightforward.

Despite the simple implementation procedure of the local MQ-DQ method, we cannot get its analytical formulas for derivative approximation. As a result, it is difficult to theoretically analyze this scheme, such as the influence of shape parameter. In addition, it is very difficult to compare this meshless method with conventional numerical methods, such as finite difference scheme. To resolve these problems, we apply the idea of local MQ-DQ method to the stencil of the central difference scheme to derive the new MQ-FD method so that it can be compared with the conventional central difference scheme. In this study, we mainly focus on the derivation of the MQ-FD method and its performance study for derivative approximation and solution of partial differential equations.

3.2 Description of MQ-FD Methods and Comparison with Central FD Schemes

In this section, the derivation of the MQ-FD methods both in 1-D space and 2-D space is presented in detail. The basic idea to derive MQ-FD method is the same as that in local MQ-DQ method, which can be found in Chapter 2. A global nodal index is used to identify points in the domain. For any reference point i, there is a

supporting region, as shown in Figure 3.1 (for 1-D space) and Figure 3.2 (for 2-D space). A local nodal index is used to identify the supporting points for the reference point.

3.2.1 MQ-FD method in 1-D space

If a function $f(x)$ is assumed to be sufficiently smooth, its first and second order derivatives with respect to x at a point x_i can be approximated by the MQ-FD method as

$$f_x^{(1)}(x_i) = \sum_{k=1}^{3} w_{i,k}^{(1)} f(x_{i,k}) \tag{3.1}$$

$$f_x^{(2)}(x_i) = \sum_{k=1}^{3} w_{i,k}^{(2)} f(x_{i,k}) \tag{3.2}$$

where $w_{i,k}^{(1)}$ and $w_{i,k}^{(2)}$ are the related weighting coefficients, which need to be determined. $x_{i,k}$ represents the position of the kth supporting point for reference point i.

As shown in Figure 3.1, in the supporting region for reference point i, function $f(x)$ can be locally approximated by MQ RBFs as

$$f(x) = \sum_{j=1}^{2} \lambda_j g_j(x) + \lambda_3 \tag{3.3}$$

where

$$g_j(x) = \sqrt{(x - x_j)^2 + c^2} - \sqrt{(x - x_3)^2 + c^2}, \qquad c > 0 \tag{3.4}$$

c is the shape parameter given by the user.

From the property of a linear vector space, if all the base functions, $g_j(x)$, (j=1,

2) and $g_3(x)=1$, satisfy the linear relationship (3.1) or (3.2), so does any function represented by equation (3.3). Thus when the weighting coefficients of DQ approximation are determined by all the base functions, they can be used to discretize the derivatives in a PDE, whose solution can actually be represented by equation (3.3).

Substituting the three base functions into equations (3.1) and (3.2), we can get a set of linear equations, which can be expressed in matrix form as

$$[D]=[G][W] \tag{3.5}$$

where $[D]=\begin{bmatrix} 0 & 0 \\ C & E \\ -C & E \end{bmatrix}$, $[G]=\begin{bmatrix} 1 & 1 & 1 \\ A & B & -A \\ B & A & -A \end{bmatrix}$ and $[W]=\begin{bmatrix} w_{i,1}^{(1)} & w_{i,1}^{(2)} \\ w_{i,2}^{(1)} & w_{i,2}^{(2)} \\ w_{i,3}^{(1)} & w_{i,3}^{(2)} \end{bmatrix}$.

$A=c-\sqrt{\Delta^2+c^2}$, $B=\sqrt{4\Delta^2+c^2}-\sqrt{\Delta^2+c^2}$, $C=\dfrac{\Delta}{\sqrt{\Delta^2+c^2}}$ and

$E=\dfrac{c^2}{(\sqrt{\Delta^2+c^2})^3}-\dfrac{1}{c}$. Δ is the mesh spacing.

Based on Cramer's rule, the elements of matrix $[W]$ can be obtained. First of all, we will illustrate the procedure of obtaining the weighting coefficients for the first order derivative. Determinants of matrices can be expressed as

$$|G|=\begin{vmatrix} 1 & 1 & 1 \\ A & B & -A \\ B & A & -A \end{vmatrix}=-2AB+3A^2-B^2=(3A+B)(A-B),$$

$$|G_{11}|=\begin{vmatrix} 0 & 1 & 1 \\ C & B & -A \\ -C & A & -A \end{vmatrix}=3AC+BC=(3A+B)C,$$

$$|G_{21}| = \begin{vmatrix} 1 & 0 & 1 \\ A & C & -A \\ B & -C & -A \end{vmatrix} = -3AC - BC = -(3A + B)C$$

and $|G_{31}| = \begin{vmatrix} 1 & 1 & 0 \\ A & B & C \\ B & A & -C \end{vmatrix} = 0$.

Thus, the weighting coefficients are

$$w_{i,1}^{(1)} = \frac{|G_{11}|}{|G|} = \frac{(3A + B)C}{(3A + B)(A - B)} = \frac{C}{A - B},$$

$$w_{i,2}^{(1)} = \frac{|G_{21}|}{|G|} = \frac{-(3A + B)C}{(3A + B)(A - B)} = \frac{-C}{A - B}$$

and $w_{i,3}^{(1)} = \frac{|G_{31}|}{|G|} = \frac{0}{(3A + B)(A - B)} = 0$.

With the above weighting coefficients, the first order derivative can be expressed as

$$\begin{aligned} f_x^{(1)}(x_i) &= \frac{C}{A - B}[f(x_{i,1}) - f(x_{i,2})] \\ &= \frac{\Delta}{(\sqrt{4\Delta^2 + c^2} - c)\sqrt{\Delta^2 + c^2}}[f(x_{i,2}) - f(x_{i,1})] \end{aligned} \tag{3.6}$$

Compared with the formula of the central FD scheme for the first order derivative, i.e., $f_x^{(1)}(x_i) = \frac{1}{2\Delta}[f(x_{i,2}) - f(x_{i,1})]$, the formula of the MQ-FD method is dependent on the value of the shape parameter *c*. In the following, we will discuss the effect of the shape parameter on the formula of the MQ-FD method.

1) When *c* goes to infinity, according to the Binomial Theorem, we have

$$\lim_{c\to\infty}\sqrt{\Delta^2 + c^2} = \lim_{c\to\infty} c(1 + \frac{\Delta^2}{c^2})^{\frac{1}{2}} = \lim_{c\to\infty} c(1 + \frac{1}{2}\frac{\Delta^2}{c^2} - \frac{1}{8}\frac{\Delta^4}{c^4} + \cdots) = \lim_{c\to\infty}(c + \cdots),$$

$$\lim_{c\to\infty}(\sqrt{4\Delta^2+c^2}-c)=\lim_{c\to\infty}[c(1+\frac{1}{2}\frac{4\Delta^2}{c^2}-\frac{1}{8}\frac{16\Delta^4}{c^4}+\cdots)-c]=\lim_{c\to\infty}(\frac{2\Delta^2}{c}+\cdots)$$ and

$$\lim_{c\to\infty}\frac{\Delta}{(\sqrt{4\Delta^2+c^2}-c)\sqrt{\Delta^2+c^2}}=\frac{\Delta}{\lim_{c\to\infty}(\sqrt{4\Delta^2+c^2}-c)\lim_{c\to\infty}\sqrt{\Delta^2+c^2}}$$

$$=\frac{\Delta}{\lim_{c\to\infty}(\frac{2\Delta^2}{c}+\cdots)\lim_{c\to\infty}(c+\cdots)}=\frac{1}{2\Delta}$$.

Thus, when *c* goes to infinity, we can get $f_x^{(1)}(x_i)=\frac{1}{2\Delta}[f(x_{i,2})-f(x_{i,1})]$, which is the same as the formula of the central FD scheme. This is consistent with the observations by Driscoll and Fornberg (2002) and Wright and Fornberg (2006). This observation shows that the "classical" polynomial-based FD scheme can be reproduced by the MQ-FD method in the limit of $c\to\infty$.

2) When $0<c<\infty$, we numerically compare the formula of the MQ-FD method with that of the central FD scheme. Dividing the formula of the MQ-FD method by that of the central FD scheme, we can get a coefficient of $\frac{2\Delta^2}{(\sqrt{4\Delta^2+c^2}-c)\sqrt{\Delta^2+c^2}}$. Figure 3.3(a) plots the curves of the coefficient according to the shape parameter *c* with Δ to be 0.04 (26 points), 0.02 (51 points) and 0.01 (101 points), respectively. From this figure, we can see that the coefficient is always larger than 1 and when *c* goes to infinity, the coefficient approaches 1, which is consistent with the above observation. Another point to be emphasized here is that if the shape parameter *c* is fixed, with decreasing Δ, the coefficient gets closer to 1. This phenomenon can also be seen from Figure 3.3(b), in which the curves of coefficient with regard to Δ are displayed for the shape parameter *c* of 0.2, 0.5 and 1, respectively.

The procedure for computing the weighting coefficients of the second order derivative is the same. Determinants of matrices are

$$|G_{12}| = \begin{vmatrix} 0 & 1 & 1 \\ E & B & -A \\ E & A & -A \end{vmatrix} = AE - BE = (A-B)E \,,$$

$$|G_{22}| = \begin{vmatrix} 1 & 0 & 1 \\ A & E & -A \\ B & E & -A \end{vmatrix} = AE - BE = (A-B)E$$

$$\text{and} |G_{32}| = \begin{vmatrix} 1 & 1 & 0 \\ A & B & E \\ B & A & E \end{vmatrix} = 2BE - 2AE = -2(A-B)E \,.$$

Thus, the weighting coefficients can be obtained as

$$w_{i,1}^{(2)} = \frac{|G_{12}|}{|G|} = \frac{(A-B)E}{(3A+B)(A-B)} = \frac{E}{3A+B}$$

$$w_{i,2}^{(2)} = \frac{|G_{22}|}{|G|} = \frac{(A-B)E}{(3A+B)(A-B)} = \frac{E}{3A+B}$$

$$w_{i,3}^{(2)} = \frac{|G_{32}|}{|G|} = \frac{-2(A-B)E}{(3A+B)(A-B)} = -2\frac{E}{3A+B}$$

With these weighting coefficients, the second order derivative can be expressed as

$$\begin{aligned} f_x^{(2)}(x_i) &= \frac{E}{3A+B}[f(x_{i,1}) + f(x_{i,2}) - 2f(x_{i,3})] \\ &= \frac{\dfrac{c^2}{(\sqrt{\Delta^2+c^2})^3} - \dfrac{1}{c}}{3c - 4\sqrt{\Delta^2+c^2} + \sqrt{4\Delta^2+c^2}}[f(x_{i,1}) + f(x_{i,2}) - 2f(x_{i,3})] \end{aligned} \tag{3.7}$$

Similar to the procedure for the first order derivative, we study the effect of the shape parameter c on the formula of MQ-FD method for the second order derivative.

1) When *c* goes to infinity,

$$\lim_{c\to\infty}[\frac{c^2}{\sqrt{\Delta^2+c^2}^3}-\frac{1}{c}]=\lim_{c\to\infty}\frac{c^3-\sqrt{\Delta^2+c^2}^3}{c\sqrt{\Delta^2+c^2}^3}$$

$$=\lim_{c\to\infty}\frac{c^3-c^3(1+\frac{3}{2}\frac{\Delta^2}{c^2}+\frac{3}{8}\frac{\Delta^4}{c^4}+\cdots)}{c^4(1+\frac{3}{2}\frac{\Delta^2}{c^2}+\frac{3}{8}\frac{\Delta^4}{c^4}+\cdots)}=\lim_{c\to\infty}(-\frac{3}{2}\frac{\Delta^2}{c^3}+\cdots),$$

$$\lim_{c\to\infty}(3c-4\sqrt{\Delta^2+c^2}+\sqrt{4\Delta^2+c^2})$$

$$=\lim_{c\to\infty}[3c-4c(1+\frac{1}{2}\frac{\Delta^2}{c^2}-\frac{1}{8}\frac{\Delta^4}{c^4}+\cdots)+c(1+\frac{1}{2}\frac{4\Delta^2}{c^2}-\frac{1}{8}\frac{16\Delta^4}{c^4}+\cdots)].$$

$$=\lim_{c\to\infty}(-\frac{3}{2}\frac{\Delta^4}{c^3}+\cdots)$$

Thus, when *c* goes to infinity, we have $f_x^{(2)}(x_i)=\frac{1}{\Delta^2}[f(x_{i,1})+f(x_{i,2})-2f(x_{i,3})]$. This formula is also the same as that by the central difference scheme, which confirms the above observation.

2) When $0<c<\infty$, we also numerically compare the formula of the MQ-FD method with that of the central FD scheme. The corresponding coefficient for the second order derivative is $\frac{\frac{c^2}{(\sqrt{\Delta^2+c^2})^3}-\frac{1}{c}}{3c-4\sqrt{\Delta^2+c^2}+\sqrt{4\Delta^2+c^2}}\Delta^2$. Figure 3.4(a) plots the curves of the coefficient versus the shape parameter *c* with Δ to be 0.04 (26 points), 0.02 (51 points) and 0.01 (101 points), respectively. Figure 3.4(b) presents the curves of coefficient versus Δ with the shape parameter *c* to be 0.2, 0.5 and 1, respectively. From these two figures, we can see that, similar to those for the first order derivatives, with Δ fixed, the larger the *c*, the closer the coefficient approaches 1 and with *c* fixed, the smaller the Δ, the closer the coefficient approaches 1.

3.2.2 MQ-FD method in 2-D space

If a function $f(x,y)$ is assumed to be sufficiently smooth, its second order derivatives with respect to *x* and *y*, at a point (x_i, y_i) can be approximated by the MQ-FD method as

$$f_x^{(2)}(x_i, y_i) = \sum_{k=1}^{5} w_{i,k}^{(2)} f(x_{i,k}, y_{i,k}) \tag{3.8}$$

$$f_y^{(2)}(x_i, y_i) = \sum_{k=1}^{5} \overline{w}_{i,k}^{(2)} f(x_{i,k}, y_{i,k}) \tag{3.9}$$

where $w_{i,k}^{(2)}$ and $\overline{w}_{i,k}^{(2)}$ are the related weighting coefficients in the *x* and *y* directions, which need to be determined. $(x_{i,k}, y_{i,k})$ represents the position of the *k*th supporting point for reference point i.

As shown in Figure 3.2, in the supporting region for reference point i, function $f(x,y)$ can be locally approximated by MQ RBFs as

$$f(x,y) = \sum_{j=1}^{4} \lambda_j g_j(x,y) + \lambda_5 \tag{3.10}$$

where

$$g_j(x,y) = \sqrt{(x-x_j)^2 + (y-y_j)^2 + c^2} - \sqrt{(x-x_5)^2 + (y-y_5)^2 + c^2} \tag{3.11}$$

Substituting the five base functions, $g_j(x,y)$, (j=1,…,4) and $g_5(x,y) = 1$, into equations (3.8) and (3.9), we can obtain a set of linear equations, which can be expressed in matrix form as

$$[D] = [G][W] \tag{3.12}$$

where $[D]=\begin{bmatrix} 0 & 0 \\ X & Y \\ Y & X \\ X & Y \\ Y & X \end{bmatrix}$, $[G]=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ A & B & C & B & -A \\ B & A & B & C & -A \\ C & B & A & B & -A \\ B & C & B & A & -A \end{bmatrix}$ and $[W]=\begin{bmatrix} w_{i,1}^{(2)} & \overline{w}_{i,1}^{(2)} \\ w_{i,2}^{(2)} & \overline{w}_{i,2}^{(2)} \\ w_{i,3}^{(2)} & \overline{w}_{i,3}^{(2)} \\ w_{i,4}^{(2)} & \overline{w}_{i,4}^{(2)} \\ w_{i,5}^{(2)} & \overline{w}_{i,5}^{(2)} \end{bmatrix}$.

$A=c-\sqrt{\Delta^2+c^2}$, $B=\sqrt{2\Delta^2+c^2}-\sqrt{\Delta^2+c^2}$, $C=\sqrt{4\Delta^2+c^2}-\sqrt{\Delta^2+c^2}$,

$X=\frac{c^2}{(\sqrt{\Delta^2+c^2})^3}-\frac{1}{c}$ and $Y=\frac{1}{\sqrt{\Delta^2+c^2}}-\frac{1}{c}$.

The procedure for coefficients in 2-D space is similar to that in 1-D space. Compared with the 3×3 dimensional matrix in 1-D space, G in 2-D space is a 5×5 dimensional matrix. Thus we cannot compute the determinants of matrices directly. Instead, the software Maple is used to compute the determinants. Determinants of matrices are:

$|G|=(C-A)^2(5A+C+2B)(A+C-2B)$

$|G_{11}|=(C-A)^2(XC+3XA-2YA-2BY)$

$|G_{21}|=(C-A)^2(YC+3YA-2XA-2XB)$

$|G_{31}|=(C-A)^2(XC+3XA-2YA-2BY)$

$|G_{41}|=(C-A)^2(YC+3YA-2XA-2XB)$

and $|G_{51}|=-2(C-A)^2(A+C-2B)(X+Y)$

Thus, the weighting coefficients can be obtained as

$$w_{i,1}^{(2)}=\frac{|G_{11}|}{|G|}=\frac{XC+3XA-2YA-2BY}{(5A+C+2B)(A+C-2B)}$$

$$w_{i,2}^{(2)}=\frac{|G_{21}|}{|G|}=\frac{YC+3YA-2XA-2XB}{(5A+C+2B)(A+C-2B)}$$

$$w_{i,3}^{(2)} = \frac{|G_{31}|}{|G|} = \frac{XC + 3XA - 2YA - 2BY}{(5A + C + 2B)(A + C - 2B)}$$

$$w_{i,4}^{(2)} = \frac{|G_{41}|}{|G|} = \frac{YC + 3YA - 2XA - 2XB}{(5A + C + 2B)(A + C - 2B)}$$

$$w_{i,5}^{(2)} = \frac{|G_{51}|}{|G|} = -2\frac{(X + Y)}{5A + C + 2B}$$

When c goes to infinity, based on the binomial theorem, we have:

$$\lim_{c\to\infty}\sqrt{\Delta^2 + c^2} = \lim_{c\to\infty}(c + \frac{1}{2}\frac{\Delta^2}{c} - \frac{1}{8}\frac{\Delta^4}{c^3} + \cdots)$$

$$\lim_{c\to\infty}\sqrt{2\Delta^2 + c^2} = \lim_{c\to\infty}(c + \frac{\Delta^2}{c} - \frac{1}{2}\frac{\Delta^4}{c^3} + \cdots)$$

$$\lim_{c\to\infty}\sqrt{4\Delta^2 + c^2} = \lim_{c\to\infty}(c + 2\frac{\Delta^2}{c} - 2\frac{\Delta^4}{c^3} + \cdots)$$

$$\lim_{c\to\infty}(\Delta^2 + c^2)^{-\frac{3}{2}} = \lim_{c\to\infty}(\frac{1}{c^3} - \frac{3}{2}\frac{\Delta^2}{c^5} + \frac{15}{8}\frac{\Delta^4}{c^7} + \cdots)$$

$$\lim_{c\to\infty}(\Delta^2 + c^2)^{-\frac{1}{2}} = \lim_{c\to\infty}(\frac{1}{c} - \frac{1}{2}\frac{\Delta^2}{c^3} + \frac{3}{8}\frac{\Delta^4}{c^5} + \cdots)$$

Thus, we can get:

$$X + Y = \lim_{c\to\infty}(-2\frac{\Delta^2}{c^3} + \cdots)$$

$$5A + C + 2B = \lim_{c\to\infty}(-2\frac{\Delta^4}{c^3} + \cdots)$$

$$A + C - 2B = \lim_{c\to\infty}(-\frac{\Delta^4}{c^3} + \cdots)$$

$$XC + 3XA - 2YA - 2BY = X(C + 3A) - 2Y(A + B) = \lim_{c\to\infty}(2\frac{\Delta^6}{c^6} + \cdots)$$

and $YC + 3YA - 2XA - 2XB = Y(C + 3A) - 2X(A + B) = 0$

Finally, the coefficients can be obtained as:

$$w_{i,1}^{(2)} = w_{i,3}^{(2)} = \frac{\lim_{c\to\infty}(2\frac{\Delta^6}{c^6}+\cdots)}{\lim_{c\to\infty}(-2\frac{\Delta^4}{c^3}+\cdots)\lim_{c\to\infty}(-\frac{\Delta^4}{c^3}+\cdots)} = \frac{1}{\Delta^2}$$

$$w_{i,2}^{(2)} = w_{i,4}^{(2)} = 0$$

and $$w_{i,5}^{(2)} = -2\frac{\lim_{c\to\infty}(-2\frac{\Delta^2}{c^3}+\cdots)}{\lim_{c\to\infty}(-2\frac{\Delta^4}{c^3}+\cdots)} = \frac{-2}{\Delta^2}$$

Thus, when c goes to infinity, we have:

$$f_x^{(2)}(x_i, y_i) = \sum_{k=1}^{5} w_{i,k}^{(2)} f(x_{i,k}, y_{i,k}) = \frac{f(x_{i,1}, y_{i,1}) + f(x_{i,3}, y_{i,3}) - 2f(x_{i,5}, y_{i,5})}{\Delta^2} \quad (3.13)$$

Based on the symmetric property, we have

$$f_y^{(2)}(x_i, y_i) = \frac{f(x_{i,2}, y_{i,2}) + f(x_{i,4}, y_{i,4}) - 2f(x_{i,5}, y_{i,5})}{\Delta^2} \quad (3.14)$$

when c goes to infinity. These formulas are the same as those by the central difference scheme, which is consistent with the above observation.

3.3 Performance Study of MQ-FD Methods for Derivative Approximation and Solution of Poisson Equations

In this section, we study the performance of the MQ-FD methods for derivative approximation and solution of Poisson equations. Derivatives in 1-D space and Poisson equations in 2-D space are taken as model problems and results are compared with those obtained by the central FD scheme.

3.3.1 Derivative approximation of the MQ-FD method in 1-D space

In this part, the first and second order derivatives of two functions, $f = \sin(\pi x)$ and $f = x^4$ are approximated by both the MQ-FD method and the central FD scheme. Accuracy obtained by the MQ-FD method with different shape parameters is shown in Figure 3.5 and Figure 3.6, in which, accuracy of the central FD scheme is also displayed for comparison. The grid is chosen to be 51×51 and the function values on the grid points are taken as known.

Figure 3.5 indicates that, as compared with the central FD scheme, the MQ-FD method may approximate the derivatives of the function $f = \sin(\pi x)$ more accurately or less accurately according to the choice of shape parameter *c*. When the value of *c* is very small, the accuracy of derivative approximation by the MQ-FD method is very low. With increase of the *c* value, there exists a range of *c* with which the MQ-FD method approximates the derivatives more accurately than the central FD scheme does. When the value of the shape parameter *c* goes to infinity, the accuracy achieved by the MQ-FD method approaches that by the central FD scheme. Comparatively, the accuracy of derivative approximation of the function $f = x^4$ achieved by the MQ-FD method is always lower than that by the central FD scheme whatever *c* is, as shown in Figure 3.6. However, when *c* goes to infinity, the accuracy by the MQ-FD method also goes to that by the central FD scheme. This is in good agreement with what we have derived in the last section that when *c* goes to infinity, the MQ-FD method can be reduced to the central FD scheme.

3.3.2 Application for solution of Poisson equations in 2-D space

In this part, we study the performance of the MQ-FD method for the solution of Poisson equation in 2-D space. Poisson equation is taken as a model problem, which can be written as:

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}=f(x,y) \quad \text{in} \quad \Omega=\{(x,y)\,|\,0\le x,y\le 1\},\ T=g \quad \text{on} \quad \partial\Omega \tag{3.15}$$

where f and g are determined in such a manner that the exact solution T of the Poisson equation is the given one.

To study the performance of the MQ-FD method in simulating two classical types of flow problems: periodic boundary value problems and general boundary value problems, we take $T=\sin(\pi x)\sin(\pi y)$ and $T=x^4+y^4$ as two typical solution functions. Here, $T=\sin(\pi x)\sin(\pi y)$ can represent the solution of the periodic boundary value problems and $T=x^4+y^4$ can stand for the solution of the general boundary value problems. First, we will observe the effect of the shape parameter c on the MQ-FD result. Accuracy obtained by the MQ-FD method with different shape parameters is shown in Figure 3.7, in which, accuracy of the central FD scheme is also displayed for comparison. The grid is chosen to be 51×51. Comparing Figures 3.5, 3.6 and 3.7, we can see that the effect of the shape parameter on the performance of the MQ-FD method for the solution of Poisson equations in 2-D space is very similar to that on its performance for derivative approximation in 1-D space.

Then we illustrate the convergence rate of the MQ-FD method as the grid is refined. To study the convergence rate of the method, we choose ten different values of the shape parameter c, ranging from 0.02 to 10, and solve the Poisson equation with grid sizes of 21×21, 41×41, 61×61, 81×81 and 101×101. Convergence rate of the MQ-FD method with different shape parameters and that of the central FD scheme are shown in Figure 3.8. From this figure, we can see that the slope of the convergence rate of the MQ-FD method with certain shape parameter *c*, say from 0.1 to 5, is parallel to that of the central FD scheme. However, when the shape parameter is very small (0.02 and 0.05) or very large (8 and 10), the symbols representing the accuracy of solution are not in a line. This is reasonable. It is well known that when *c* is very small, the MQ-FD method can not solve the Poisson equation accurately and when *c* is very large, the condition number of matrix G in equation (3.12) becomes very large. That is, matrix G becomes highly ill-conditioned, leading to a large numerical error of MQ-FD method. Another point to be emphasized here is that for $T = \sin(\pi x)\sin(\pi y)$, with the increase of the shape parameter *c*, the accuracy of the MQ-FD method can be improved to be higher than that of the central FD scheme. However, with further increase of the shape parameter *c*, the accuracy of the MQ-FD method will be decreased due to ill-condition of the MQ-FD matrices. For $T = x^4 + y^4$, with the increase of the shape parameter *c*, the accuracy of the MQ-FD method can be improved to approach that of the central FD scheme. These results are consistent

with those in Figure 3.7.

3.4 Simulation of Lid-driven Flow in a Square Cavity

In the last section, we have carried out the performance study of the MQ-FD methods for derivative approximation and solution of Poisson equations. Now, we will illustrate the ability of the MQ-FD method for solving fluid flow problems accurately. In this study, a steady incompressible lid-driven flow in a square cavity is taken as a model problem, as shown schematically in Figure 3.9. The governing equations are the two dimensional steady incompressible Navier-Stokes equations in the vorticity-stream function form, which can be written as

$$u\frac{\partial \omega}{\partial x}+v\frac{\partial \omega}{\partial y}=\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 \omega}{\partial x^2}+\frac{\partial^2 \omega}{\partial y^2}\right) \tag{3.16}$$

$$\frac{\partial^2 \psi}{\partial x^2}+\frac{\partial^2 \psi}{\partial y^2}=\omega \tag{3.17}$$

where ω is the vorticity, ψ is the stream function, *u*, *v* denote the components of velocity in the *x* and *y* directions, which can be calculated from the stream function

$$u=\frac{\partial \psi}{\partial y},\quad v=-\frac{\partial \psi}{\partial x} \tag{3.18}$$

Re is the Reynolds number. For this model problem, *Re* is chosen to be 1000 and 5000, respectively.

In this study, the governing equations (3.16)-(3.18) are discretized by the MQ-FD method. The discretization form of the governing equations at a general node *i* can be written as:

$$u_i \sum_{k=1}^{5} w_{i,k}^{(1)} \omega_i^k + v_i \sum_{k=1}^{5} \overline{w}_{i,k}^{(1)} \omega_i^k = \frac{1}{\mathrm{Re}} \left(\sum_{k=1}^{5} w_{i,k}^{(2)} \omega_i^k + \sum_{k=1}^{5} \overline{w}_{i,k}^{(2)} \omega_i^k \right) \tag{3.19}$$

$$\sum_{k=1}^{5} w_{i,k}^{(2)} \psi_i^k + \sum_{k=1}^{5} \overline{w}_{i,k}^{(2)} \psi_i^k = \omega_i \tag{3.20}$$

$$u_i = \sum_{k=1}^{5} \overline{w}_{i,k}^{(1)} \psi_i^k \,, \quad v_i = -\sum_{k=1}^{5} w_{i,k}^{(1)} \psi_i^k \tag{3.21}$$

The boundary conditions of this problem can be written as:

$u = 0$, $v = 0$, $\psi = 0$ at $x = 0,1$, $0 \le y < 1$

$u = 0$, $v = 0$, $\psi = 0$ at $y = 0$, $0 \le x \le 1$

$u = 1$, $v = 0$, $\psi = 0$ at $y = 1$, $0 \le x \le 1$

The boundary condition for ω can be derived from equation (3.17), i.e.,

$$\omega\big|_{wall} = \left. \frac{\partial^2 \psi}{\partial x^2} \right|_{wall} + \left. \frac{\partial^2 \psi}{\partial y^2} \right|_{wall} \tag{3.22}$$

The general solution procedure of the MQ-FD method for the above governing equations is shown below:

(1) Set up the node distribution (uniform Cartesian grid points) in the domain.

(2) Determine the supporting points for each reference point.

(3) Calculate the weighting coefficients for the related derivatives in the governing equations. Actually it is only necessary to compute the coefficients once and apply them all over the discretization.

(4) Discretize the governing equations with the computed weighting coefficients.

(5) Solve the resultant algebraic equations.

Firstly, the problem is solved by the MQ-FD method with a Cartesian mesh of 101×101 for $\mathrm{Re}=1000$. The shape parameter c is chosen to be 0.03 in this case. After discretizing the governing equations on all the interior points by the MQ-FD method, we get a set of linear algebraic equations. To solve the resultant equations, the successive over-relaxation (SOR) method is used. The computed velocity component u at the vertical centerline of $x=0.5$ and v at the horizontal centerline of $y=0.5$ are plotted in Figures 3.10 and 3.11. Since there is no analytical solution for this problem, the result of Ghia et al. (1982) is adopted as the benchmark data to validate the present results. The numerical results by the central FD scheme with the same grid are also plotted in the figure for comparison. In order to see the accuracy difference of these two methods clearly, only the enlarged view of the velocity component u around $y=0.2\,\&\,0.9$ and v around $x=0.2\,\&\,0.9$ is presented, where the biggest differences occur. From these figures, we can see that although both methods solve the problem very accurately, the numerical results of the MQ-FD method agree better with the benchmark solution. This means that if a proper shape parameter c is chosen, the MQ-FD method may simulate this fluid flow problem more accurately than the central FD scheme with the same Cartesian mesh. The streamlines and vorticity contours of this case by the MQ-FD method on the uniform mesh of 101×101 are shown in Figure 3.12. They also agree well with those in the work of Ghia et al. (1982).

For $\mathrm{Re} = 5000$, the Cartesian mesh is chosen to be 201×201 and the shape parameter is taken as 0.02. It is well known that as compared with the case of $\mathrm{Re} = 1000$, the flow with $\mathrm{Re} = 5000$ is much more difficult to be simulated. The computed velocity component *u* at the vertical centerline of $x = 0.5$ and *v* at the horizontal centerline of $y = 0.5$ are plotted in Figures 3.13 and 3.14, together with the results of Ghia et al. (1982) and those by the central FD scheme. From these figures, we can also observe that the results by the MQ-FD method agree better with the benchmark solution as compared with those of the central FD scheme. Figure 3.15 shows the streamlines and vorticity contours of this case. To display the secondary vortices clearly, an enlarged view of the left bottom corner is also included. These results, including the configuration and positions of the vortices, are in good agreement with those of Ghia et al. (1982). This implies that the MQ-FD method can simulate incompressible fluid flows with high Reynolds numbers accurately.

3.5 Conclusions

In this chapter, the MQ-FD method was derived and its performance for derivative approximation and solution of Poisson equation and incompressible Navier-Stokes equations was investigated. In addition, the effect of the shape parameter *c* on the formulas and accuracy of the MQ-FD method was analyzed. It was found that when *c* goes to infinity, the MQ-FD formulas of derivative approximation are the same as those given by the central FD scheme. With regard to the accuracy of the

MQ-FD methods, it was found that if the shape parameter c is properly chosen, the MQ-FD method may solve periodic boundary value problems more accurately than the central FD scheme does. For general boundary value problems, however, the accuracy by the MQ-FD method may not be as accurate as that by the central FD scheme. When the value of c is not very small, the accuracy by these two methods is very close. The lid-driven flow in a square cavity was simulated by the MQ-FD method. Results showed that with a proper shape parameter c, the MQ-FD method can simulate this flow problem very accurately, as compared with the central FD scheme.

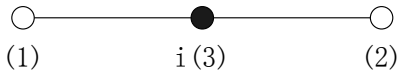


Figure 3.1 A supporting region for point i in 1-D space

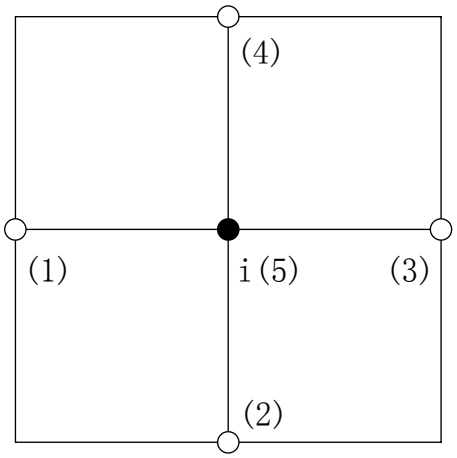
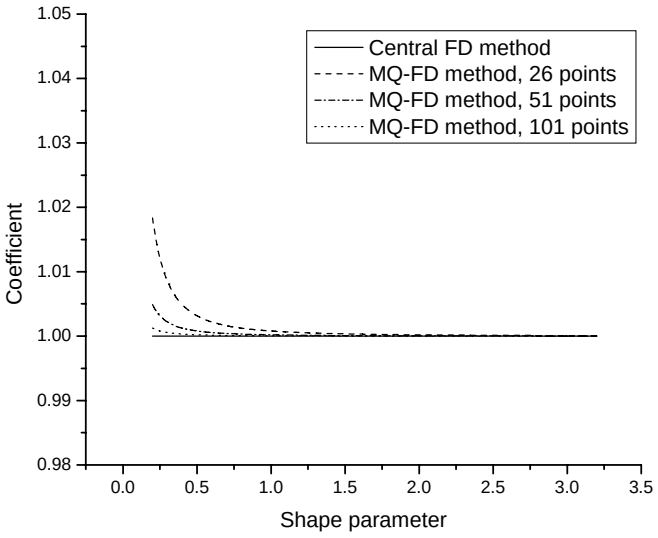
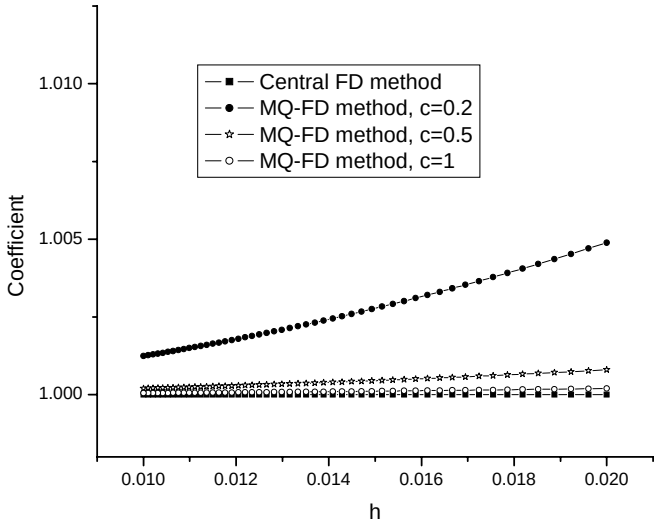


Figure 3.2 A supporting region for point i in 2-D space

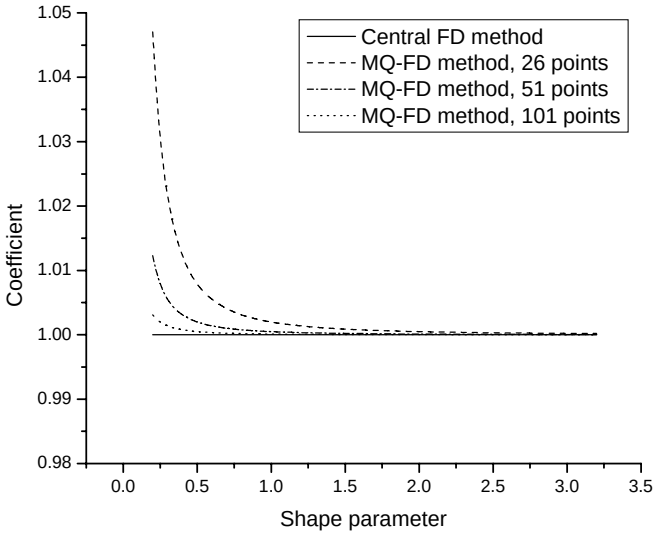


(a) Coefficient variation with regard to shape parameter *c*

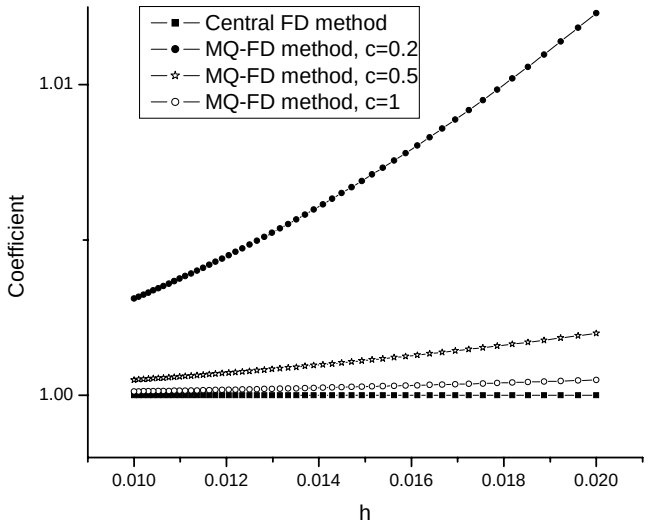


(b) Coefficient variation with regard to mesh spacing *h*

Figure 3.3 Effect of shape parameter *c* and mesh spacing *h* on the coefficient of formula for first order derivatives

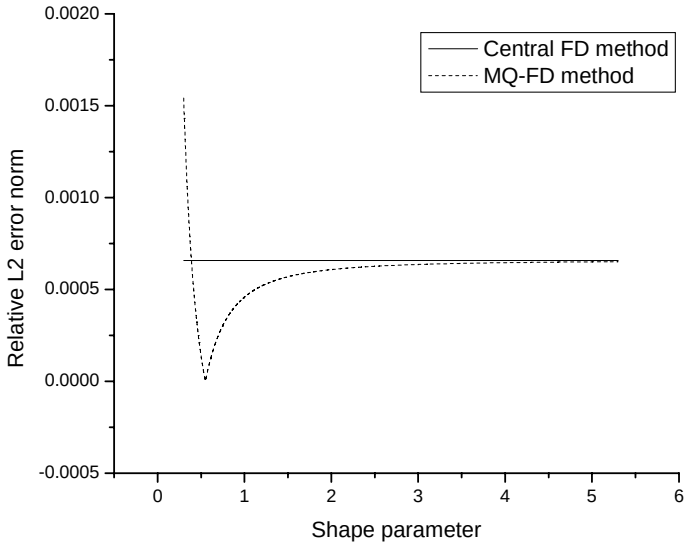


(a) Coefficient variation with regard to shape parameter *c*

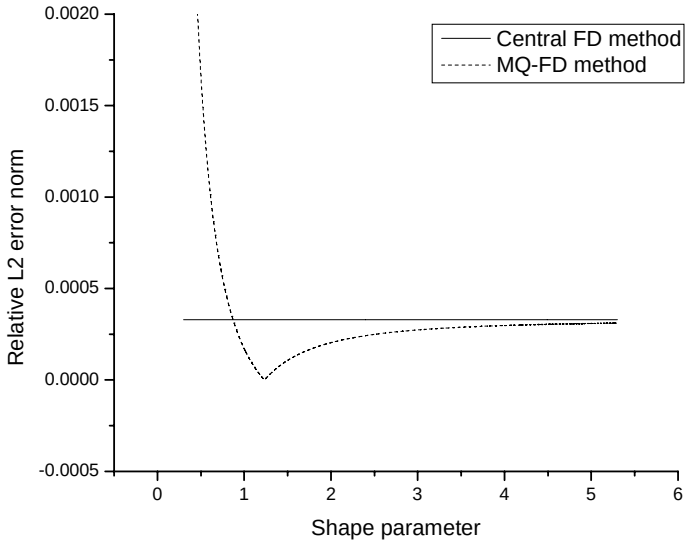


(b) Coefficient variation with regard to mesh spacing *h*

Figure 3.4 Effect of shape parameter *c* and mesh spacing *h* on the coefficient of formula for second order derivatives

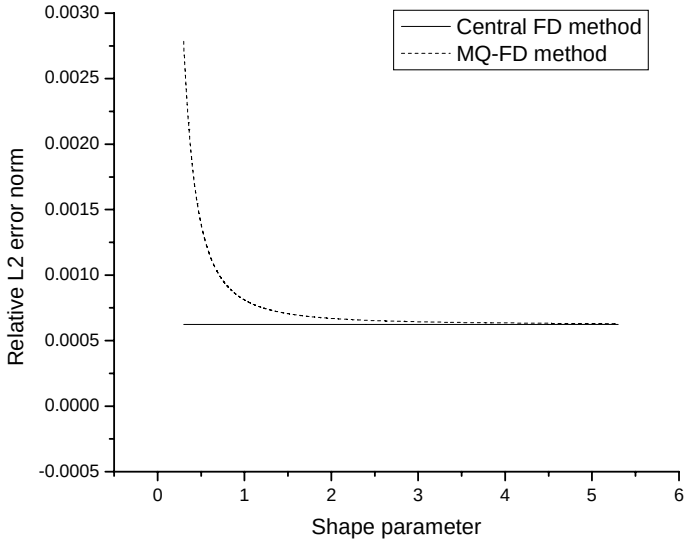


(a) First order derivative

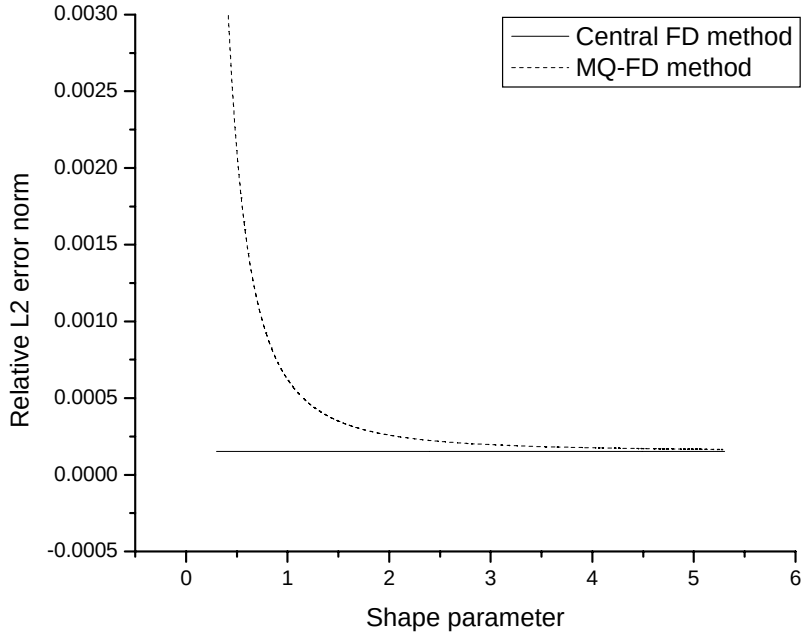


(b) Second order derivative

Figure 3.5 Derivative approximation of $\sin(\pi x)$ by the central FD method and the MQ-FD method

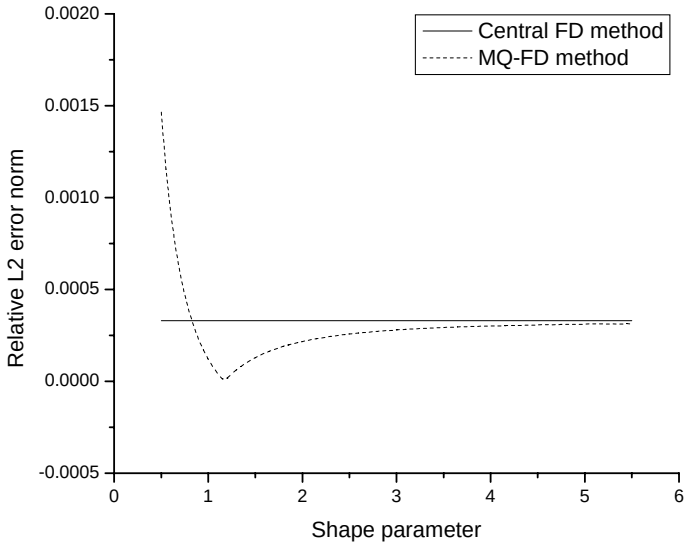


(a) First order derivative

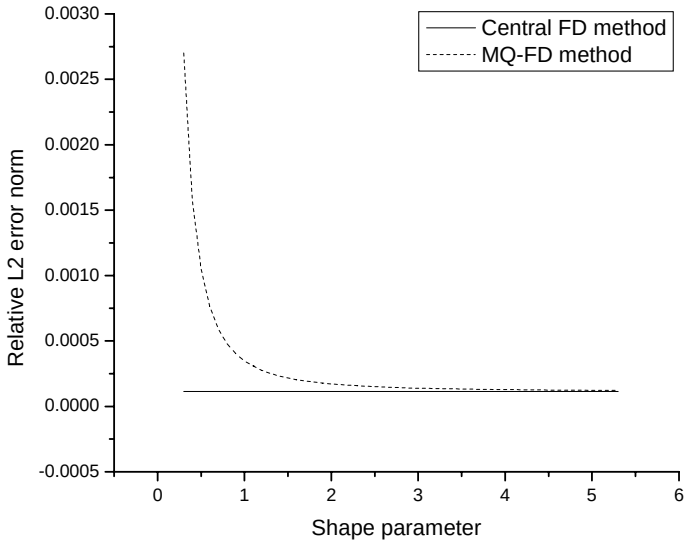


(b) Second order derivative

Figure 3.6 Derivative approximation of x^4 by the central FD method and the MQ-FD method

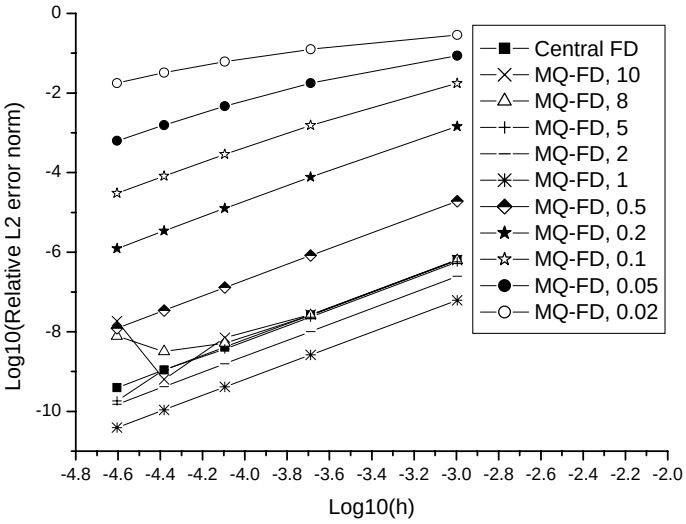


(a) $\sin(\pi x)\sin(\pi y)$

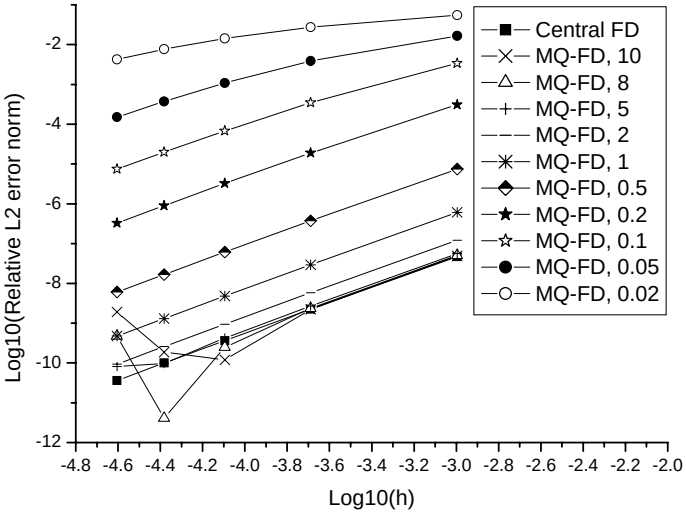


(b) $x^4 + y^4$

Figure 3.7 Comparison of accuracy between the MQ-FD method and the central FD method for solution of Poisson equations



(a) $T = \sin(\pi x)\sin(\pi y)$



(b) $T = x^4 + y^4$

Figure 3.8 Convergence rate of the MQ-FD methods with different shape parameters for solution of Poisson equation

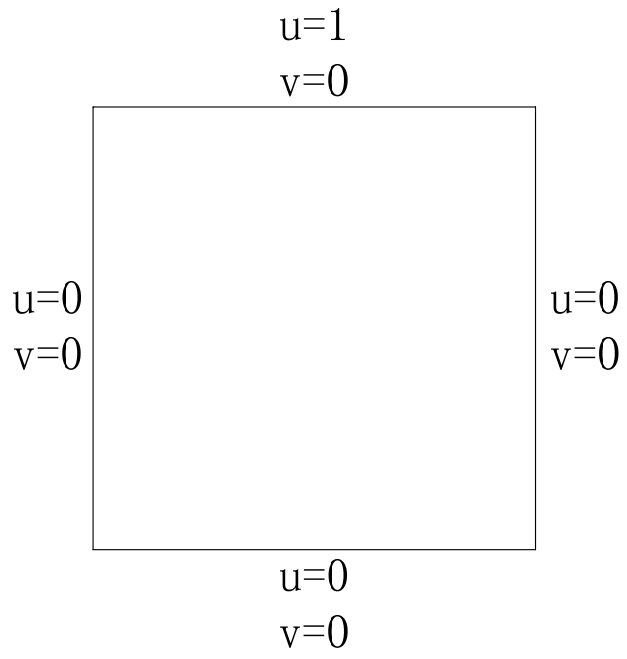
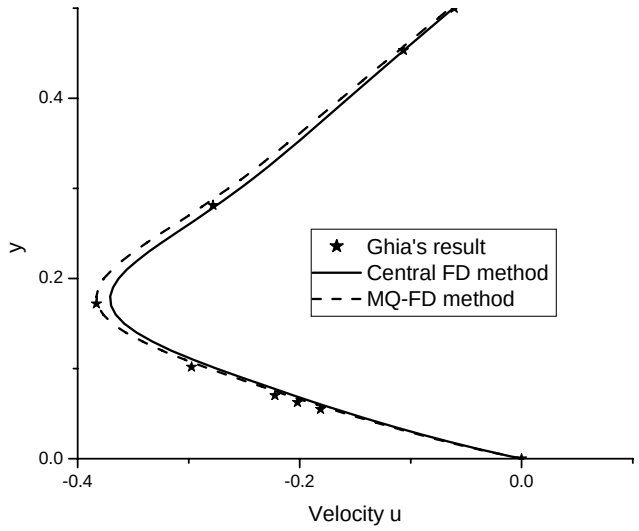
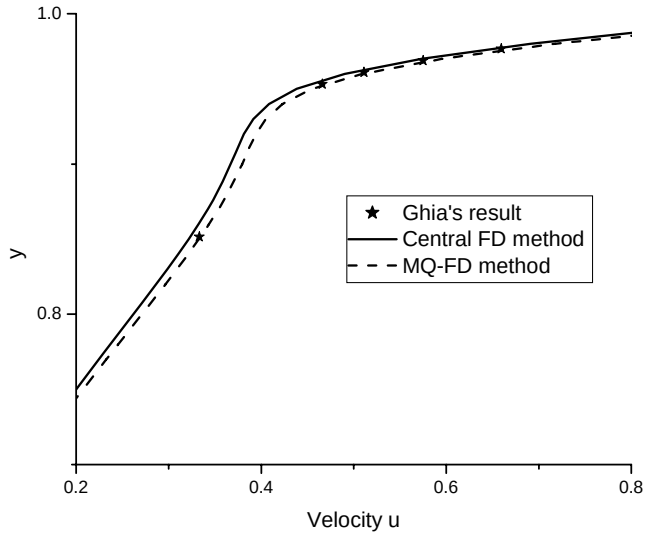


Figure 3.9 Configuration of a lid-driven flow in a square cavity

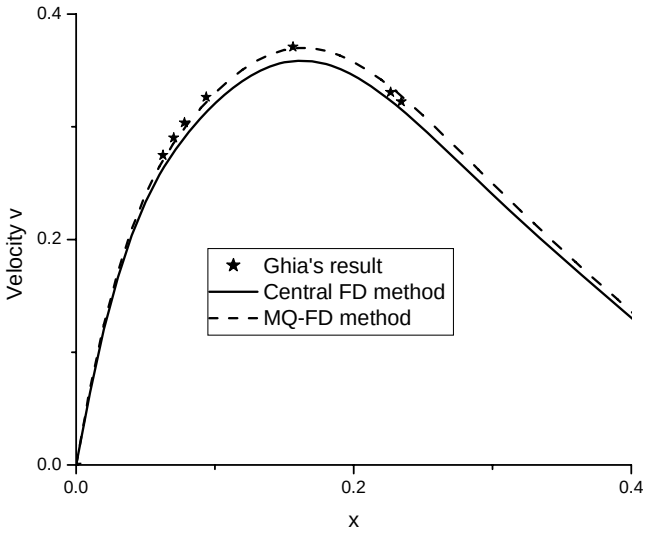


(a) Enlarged view around $y = 0.2$

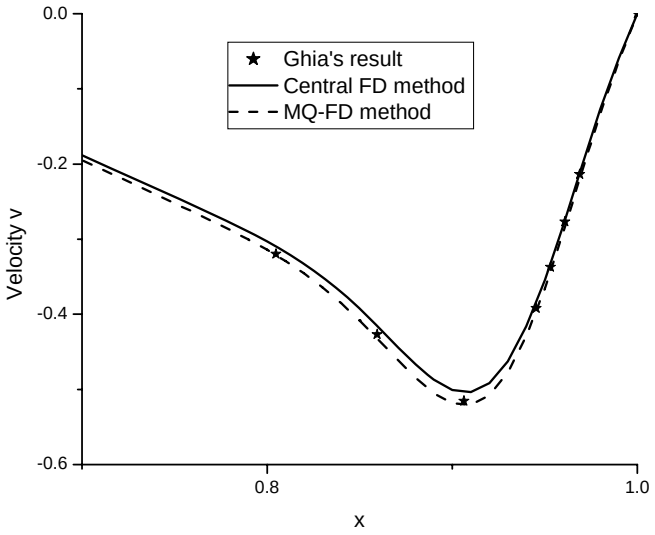


(b) Enlarged view around $y = 0.9$

Figure 3.10 Local *u*-velocity profile along vertical centerline at $\mathrm{Re} = 1000$

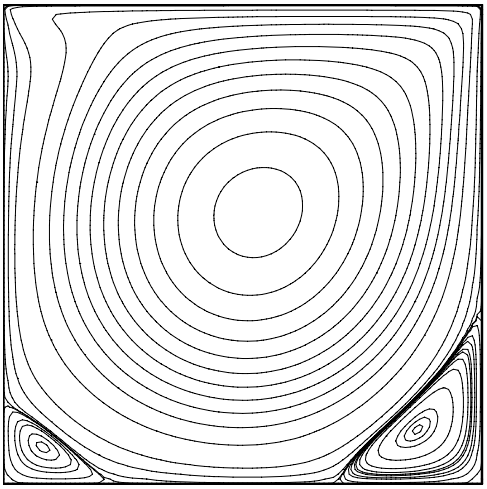


(a) Enlarged view around $x = 0.2$

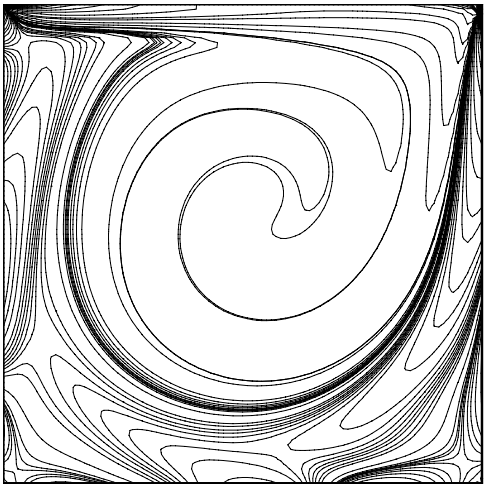


(b) Enlarged view around $x = 0.9$

Figure 3.11 Local *v*-velocity profile along horizontal centerline at $\mathrm{Re} = 1000$

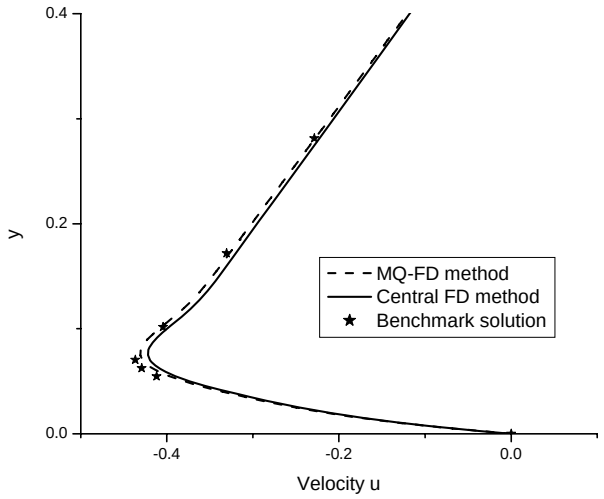


(a) Streamlines

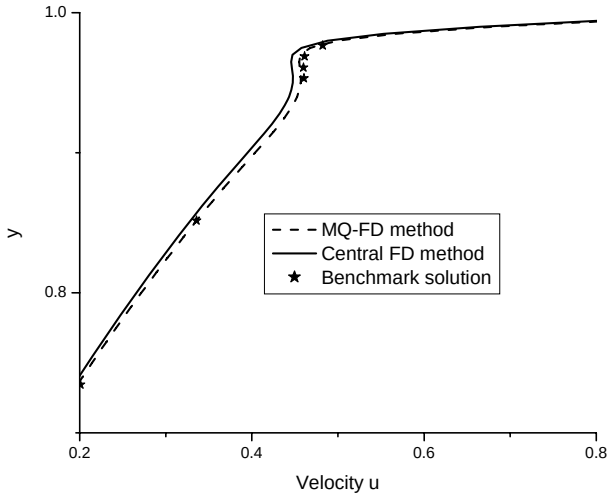


(b) Vorticity contour

Figure 3.12 Contours of lid-driven cavity flow at $Re = 1000$

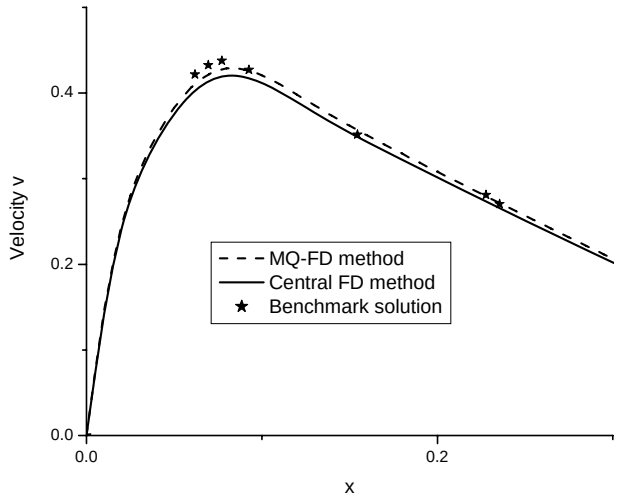


(a) Enlarged view around $y = 0.2$

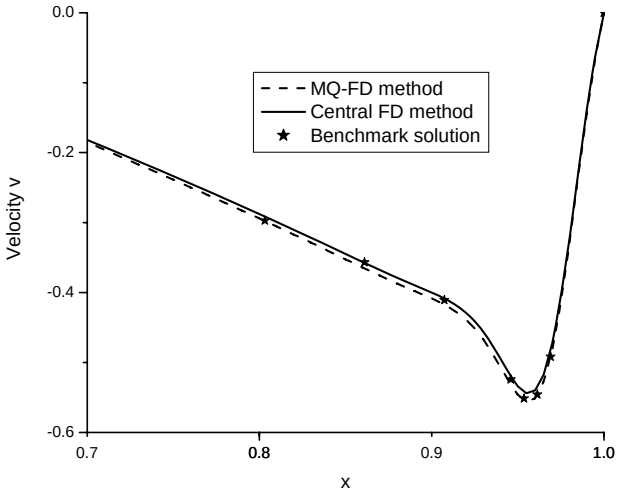


(b) Enlarged view around $y = 0.9$

Figure 3.13 Local *u*-velocity profile along vertical centerline at $\mathrm{Re} = 5000$

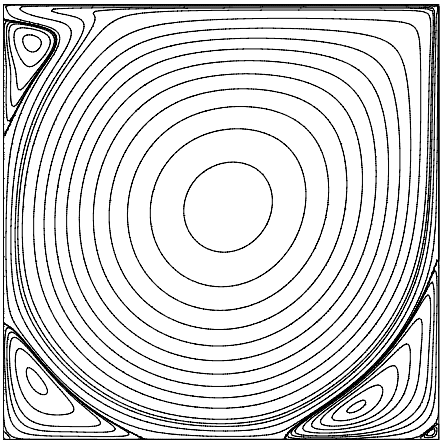


(a) Enlarged view around $x = 0.2$

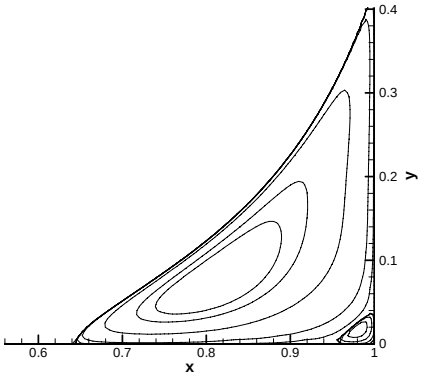


(b) Enlarged view around $x = 0.9$

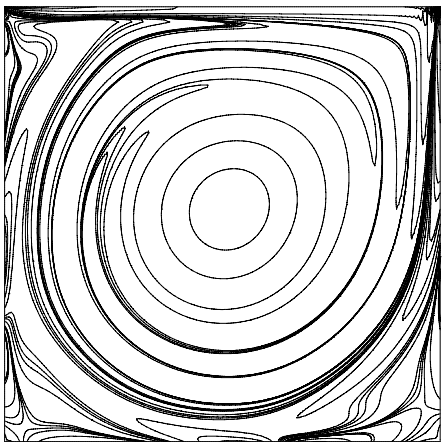
Figure 3.14 Local *v*-velocity profile along horizontal centerline at $\mathrm{Re} = 5000$



(a) Streamlines



(b) Enlarged view of the left bottom corner



(c) Vorticity contour

Figure 3.15 Contours of lid-driven cavity flow at Re = 5000

Chapter 4 Local MQ-DQ based Stencil Adaptive Method and Its Application

In this chapter, a local stencil adaptive method is presented, which is designed for solving computational fluid dynamics (CFD) problems with curved boundaries accurately. The local multiquadric-differential quadrature (MQ-DQ) method is used to discretize the governing equations, taking advantage of its meshless property. The present method bears the properties of both local MQ-DQ method for numerical discretization and local stencil adaptive method for improving computational efficiency and is thus named the local MQ-DQ based stencil adaptive method. Two test problems with curved boundaries are solved to investigate the performance of this solution-adaptive method. The numerical results indicate that the proposed method is effective and efficient by combining the advantages of meshless property for complex geometries and local adaptation for accuracy and efficiency improvement.

4.1 Motivation of this work

It is well known that one key advantage of mesh-free methods is the computational ease of adding and subtracting nodes from the existing nodes. This property is very important in solving flow problems with complicated domains or with moving boundaries. For these problems, we do not know the properties of their solutions beforehand or the changes of the solutions with time. Thus, we

cannot generate an appropriate fixed grid *a priori*. With the addition of nodes into regions where solutions are varying rapidly and the subtraction of nodes from regions where solutions are less active, these problems can be solved both accurately and efficiently. However, how to add and subtract nodes from the existing nodes is a problem for the mesh-free methods. Not much work has been done on this issue up to now. Actually, the procedure of adding and subtracting nodes from the existing nodes is under the concept of the so-called adaptive mesh refinement (AMR) techniques. In this chapter, we propose an adaptive mesh refinement technique for the local MQ-DQ method. With this adaptive mesh refinement technique, nodes will be added or subtracted based on the solutions of the problems. Thus, the local MQ-DQ method will be able to solve problems more efficiently and accurately.

4.2 Adaptive mesh refinement techniques

4.2.1 Literature review

Considerable attention has been devoted in the past few decades to the development of adaptive refinement procedures and a large number of literature were published. There are two basic approaches of grid adaptation: one is the so-called moving mesh method and the other is the mesh refinement method. Moving mesh methods generally use a fixed number of mesh points, and the points are continuously relocated as time evolves so that, at any instant time, the spatial density of the mesh points is proportional in some sense to the solution

variation or the local solution errors estimated. Several methods have been developed to determine mesh movement. The moving finite element method of Miller and Miller (1981), for example, determines the mesh movement by minimizing the residuals of the governing PDEs. A review of solution-adaptive methods of the moving mesh approach has been reported by Eiseman (1987). Further applications of the moving mesh methods can be found in Budd et al. (2005), Basebi and Thomas (2003) and Backett et al. (2002). Compared with moving mesh methods, mesh refinement methods are more popular in solving practical problems. In mesh refinement methods, the number of mesh points used is not fixed. During the computational procedure, additional points will be inserted into regions where solutions are varying rapidly and points will be deleted from regions where solutions are less active. A lot of work has been done on the development of local mesh refinement methods, see for example Berger and Oliger (1984), Berger and Leveque (1998), Zienkiewicz and Zhu (1990) and Ding and Shu (2006), with their applications reported in Howell and Greenough (2003), Hornung and Trangenstien (1997) and Anderson et al. (2004).

4.2.2 An efficient stencil adaptive algorithm

Recently, Ding and Shu (2006) proposed an efficient solution-adaptive procedure for two-dimensional incompressible viscous flows. The key idea of their stencil adaptive algorithm is to build up an adaptive hierarchy of symmetric 5-point stencils in the domain, so that central differencing can be constructed at each

interior node. The method combines the advantages of the central difference method and an adaptive mesh method in a very smart way and is therefore very effective and efficient in solving incompressible viscous flows. Despite its advantages, however, the method as proposed cannot be used to solve problems with curved boundaries directly due to the use of the central difference method. For problems with curved boundaries, before using the adaptive method, we have to implement the so-called coordinate transformation technique to map the physical domain into the computational domain. Furthermore, the original governing equations in the physical domain should also be transformed to the forms in the computational domain. The transformation procedure is usually trivial and the transformed governing equations can be very complicated. In addition, further numerical errors will be brought into the discretization process due to transformation procedure. Hence the accuracy of the solutions will be degraded as compared with the original physical domain approach.

Although the above stencil adaptive algorithm cannot be applied to solve problems with curved boundaries directly due to the use of the central difference method, its procedure for inserting and subtracting nodes from the existing nodes inspires us to couple it with the local MQ-DQ method. In this study, we propose a new stencil adaptive algorithm for flow problems with curved boundaries. The local MQ-DQ method is applied as the solver.

4.3 Development of a local MQ-DQ based stencil adaptive method

In this section, we will present in detail the development of a local MQ-DQ based stencil adaptive method. To demonstrate the procedure clearly, we will describe the method in a step-by-step manner.

4.3.1 Finite difference based stencil adaptive algorithm

The details of the finite difference based stencil adaptive algorithm can be found in the work of Ding and Shu (2006). This algorithm is based on local stencil refinement and coarsening. For any interior point in the domain, there is a local stencil associated with it. In general, there are two types of stencils encountered in this adaptive algorithm, as shown in Figure 4.1. For the convenience of inserting and deleting nodes from the adaptive stencils, only one index is used to identify the node in the domain, i.e., a global nodal index. For an arbitrary reference node i, its stencil can be symbolized as i_n^m and the position of the nodes in the stencil are denoted by X_n^m, where the superscript m denotes the resolution level, and the subscript n=0, 1, ..., 4 denotes the local index of the member nodes in the stencil.

The stencil adaptation procedure is shown below. Initially, it is easy to construct the stencil of "A" configuration around an arbitrary interior node i, as shown in Figure 4.2. For the convenience of illustration, we will take node i and its attached stencil as an example to demonstrate the whole procedure.

4.3.1.1 Criteria for stencil refinement/coarsening

First of all, we have to determine whether the stencil should be refined or coarsened. In this procedure, an action indicator is adopted to monitor the variation of the parameter of interest and send the commands to perform the corresponding process. Ding and Shu (2006) proposed two monitor parameters to measure the local variation of the solution. They are defined as follows:

1). Absolute difference, which is defined by $\Delta_1 = \max(u_i) - \min(u_i)$

2). Relative difference, which is defined by $\Delta_2 = \dfrac{\max(u_i) - \min(u_i)}{\max|u_i|}$

where u is the parameter of interest, and the subscript $i = 0,1,...,4$ denotes the local index of nodes in one stencil.

The action indicator is constructed by two thresholds. One is the upper bound $\theta_{\max}$ and the other is the lower bound $\theta_{\min}$. If $\theta_{\min} < \Delta < \theta_{\max}$, the magnitude of local variation with respect to the parameter of interest is in the range given *a priori* and no adaptation procedure will be implemented on the stencil. Otherwise, if $\Delta > \theta_{\max}$, the local variation of the parameter is relatively large and the stencil needs to be refined to improve the accuracy. Similarly, if $\Delta < \theta_{\min}$, the local variation of the parameter is relatively small and the stencil needs to be coarsened to improve the efficiency. To simplify the adaptation procedure, some constraints are introduced and checked before the process of stencil refinement/coarsening, which can be found in the work of Ding and Shu (2006). From the above, whether the stencil should be refined or coarsened has been determined. In the following,

we will show the stencil refinement and coarsening algorithms.

4.3.1.2 Stencil refinement algorithm

If stencil refinement at node i is determined by the local resolution requirement, the stencil resolution level of node i needs to be advanced from level 0 (i_n^0) to level 1 (i_n^1). Stencil refinement is achieved by the injection of extra grid points in the old stencil region to form a new stencil. In this case four new nodes are inserted. The positions of the four newly-generated nodes are actually the midpoints of the stencil edges, as shown in Figure 4.3. More specifically, they yield

$$X_1^1 = \frac{X_1^0 + X_4^0}{2},\quad X_2^1 = \frac{X_4^0 + X_3^0}{2},\quad X_3^1 = \frac{X_3^0 + X_2^0}{2} \text{ and } X_4^1 = \frac{X_2^0 + X_1^0}{2}. \quad (4.1)$$

It can be clearly seen that after the refinement, the reference node remains at the center of the stencil. Furthermore, the stencil for node i naturally evolves from the "A" configuration of i_n^0 into "B" configuration of i_n^1. It is very interesting to observe the stencils for the newly added nodes. Take the member node i_1^1 of node i as an example. From Figure 4.3, we can see that the stencil for node i_1^1 falls into type "B" category, and it also has the same stencil size as the refined stencil of the node i at resolution level 1. In other words, node i_1^1 generated during the level 1 adaptation also possesses the stencil of that resolution level. This is also the case for the other newly inserted points.

If further stencil refinement at node i is required, the stencil resolution level of

node i needs to be advanced from level 1 (i_n^1) to level 2 (i_n^2). As a consequence, four more nodes are inserted in the domain. The positions of the four nodes are also the midpoints of the stencil edges, as shown in Figure 4.4. More specifically, they yield

$$X_1^2 = \frac{X_1^1 + X_4^1}{2}, \quad X_2^2 = \frac{X_4^1 + X_3^1}{2}, \quad X_3^2 = \frac{X_3^1 + X_2^1}{2} \quad \text{and} \quad X_4^2 = \frac{X_2^1 + X_1^1}{2}. \quad (4.2)$$

It can be clearly seen from Figure 4.4 that the stencil of level 2 has the "A" configuration. We also observe that the newly added nodes also possess level 2 stencils.

If we follow the same procedure and carry on the adaptation, we can find that the stencil of the even resolution levels has the "A" configuration while the stencil of the odd resolution levels has the "B" configuration. With the adaptation continuing, the two types of stencil appear alternatively. It should be pointed out that during the stencil refinement, only the stencils of the reference node and its stencil points are affected or changed. The stencils of other nodes in the domain remain the same.

4.3.1.3 Local stencil coarsening

If a node stencil is required to be coarsened, its stencil will be recovered to the configuration of a stencil at a coarser level. As compared with the stencil refinement, the process of stencil coarsening is much easier since the adaptation information at every interior node has been recorded during previous stencil

refinement. Note that the coarsening procedure can only be carried out on resolution levels higher than the initial one, i.e., m>0.

4.3.2 Local MQ-DQ based stencil adaptive method

Although the finite difference based stencil adaptive method is very effective and efficient in solving problems with regular domains, it is not applicable in solving problems with curved boundaries. Hence the application of the method is severely limited. Here, we extend the capability of the local stencil adaptive algorithm to solve problems with curved boundaries.

For problems with curved boundaries, the background mesh will normally be non-uniform. Therefore the initial stencils for the interior nodes will not be geometrically symmetric, as shown in Figure 4.5. If stencil refinement at node i is required, similar to the process in the finite difference based stencil adaptive method, extra four grid points will be injected to form a new stencil. Before the injection of each of the four points, however, we will first search the four member nodes of the stencil of that point. Fortunately, it is not difficult due to our stencil design. Here, we will take the injection of i_1^1 as an example to illustrate the process, as shown in Figure 4.6. From Figure 4.6, we can see that three of the four member nodes of stencil i_1^1 can be immediately determined since they are actually among the old stencil i_n^0 for node i. The only exception is the member node $(i_1^1)_1^1$. However, we notice that it is also located in the stencil for nodes i_1^0

and i_4^0 of resolution level 0. Therefore, it can be accessed by either $(i_1^0)_4^0$ or $(i_4^0)_1^0$. After finding the four member nodes, the position of node i_1^1 can be easily determined. It is reasonable to insert the point at the midpoint of the four member nodes. More specifically, it yields

$$X_1^1 = \frac{(X_1^1)_1^1 + (X_1^1)_2^1 + (X_1^1)_3^1 + (X_1^1)_4^1}{4}. \tag{4.3}$$

Now the first grid point i_1^1 has been inserted and its stencil has been determined. The same process applies for the injection of other grid points. When all the four grid points are inserted, the new stencil of resolution 1 for node i has been formed, as shown in Figure 4.7.

As mentioned above, when we solve problems in rectangular geometries, the central difference method is used to approximate the spatial derivatives because each stencil is a 5-point symmetric stencil. In problems with curved boundaries, however, stencils are not geometrically symmetric and the central difference method cannot be directly used without sacrificing the accuracy. Thus, we have to look for another way to approximate the derivatives. In this stencil adaptive algorithm, for a reference node at which derivatives need to be approximated, what we have is only the information about the positions of the member nodes of the stencil. Fortunately, we can use the so-called meshless methods to approximate the derivatives with high accuracy and here the local MQ-DQ method is the selected one.

With only four supporting points, the local MQ-DQ method may not approximate derivatives very accurately. To improve the accuracy of the method, more points should be used in derivative approximation. In our work, the number of supporting points in derivative approximation is chosen to be eight, for accuracy comparable or superior to the central difference scheme on a regular Cartesian grid, based on our experience. We will show in detail how to efficiently search for the eight supporting points. For the convenience of illustration, a one-dimensional array, which is shown in Figure 4.8, will be assigned to each point to store the eight supporting points used in the local MQ-DQ approximation.

In the array, the first four supporting points, from i_1 to i_4, will be the member points of the stencil for reference point i. Comparatively, the search of the other four supporting points, from i_5 to i_8, is not so easy. Now, we will show our method of searching supporting points through the refinement procedure of an initial grid point i with the help of Figure 4.7.

Initially, it is convenient for us to search eight supporting points for point i, as shown in Figure 4.9. When the refinement procedure is implemented on point i, the stencil of this point will be advanced from resolution level 0 to resolution level 1 and four new points will be generated, i.e., from i_1^1 to i_4^1. Consequently, the eight supporting points for point i will be those shown in Figure 4.10. That is, the four member points of the stencil at resolution level 1 will be used as the first

four supporting points and the four member points of the stencil at resolution level 0 will be chosen as the last four supporting points for derivative approximation.

As for the newly generated points, we will take point i_1^1 as an example to illustrate the way of searching its eight supporting points. For this point, the first four supporting points will also be the member points of its attached stencil and they are $(i_4^0)_1^0$, i_4^0, i_0^1 and i_1^0, respectively. For each of the four member points of the stencil for point i_1^1, there is a local stencil associated with it. The last four supporting points for i_1^1 can be chosen from the member points of these four stencils. When refinement procedure is implemented on point i_1^1, the member points of the new stencil will become the first four supporting points and the member points of the old stencil will become the last four supporting points for derivative approximation. As the adaptation continues, the way of searching supporting points will be carried out in the same manner for the refined resolution levels.

4.4 Numerical Experiments

In this section, two examples are presented to examine the performance of the local MQ-DQ based stencil adaptive method. In the first test case, we investigate the accuracy of numerical solution against an analytical solution for a boundary value problem governed by the Poisson equation. For the second case, we numerically simulate a natural convective heat transfer problem in a concentric

annulus between a square outer cylinder and a circular inner cylinder.

4.4.1 Comparison with analytical solution of the Poisson equation

To illustrate the capability of the method for problems with curved boundaries, we take the computational domain to be an annulus between an inner airfoil and an outer circular cylinder. The airfoil is chosen to be a modified NACA series airfoil with profile given by

$$y(x) = \pm 5bt[0.2969\sqrt{x_{\text{int}} x} - 0.126 x_{\text{int}} x - 0.3516(x_{\text{int}} x)^2 + 0.2843(x_{\text{int}} x)^3 - 0.1015(x_{\text{int}} x)^4]$$

where $x_{\text{int}} = 1.0089$. The "+" sign is used for the upper surface and the "-" sign is used for the lower surface of the airfoil. b is the chord of the airfoil and t represents the maximum thickness. In our work, we set $b = 1$ and $t = 0.12$. The radius of the outer circular cylinder is chosen to be the same as the chord of the airfoil, i.e. $r = b$. The computational domain and the background mesh are shown in Figure 4.11.

We study the solution of the Poisson equation with Dirichlet boundary conditions on all boundaries:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = f(x, y) \tag{4.4}$$

that is, $T_B(x, y)$ is given at the domain boundaries.

For our numerical experiments, we can easily devise an analytical solution $T = \sin(\pi x)\sin(\pi y)$ for the above problem with a source term

$$f(x,y) = -2\pi^2 \sin(\pi x)\sin(\pi y) \tag{4.5}$$

and a boundary condition satisfied by $T = \sin(\pi x)\sin(\pi y)$ at the domain boundaries.

To generate a background mesh, we used a body-fitted grid generation technique. The background grid is a 40×10 mesh, as shown in Figure 4.11. After discretizing Equation (4.4) on all the interior points by the local MQ-DQ method, we get a set of linear algebraic equations. To solve the resultant equations, the Gauss-Seidel iterative method is used. In the iterative solution process, the dependent variable T at the interior points is initially set to zero. The convergence criterion is set to 10^{-10}, which is considered to be small enough for a converged solution. When the solution is converged, its accuracy is measured by the following L_2 error norm:

$$\sqrt{\frac{1}{S}\sum_{i=1}^{N}(\left|T_{i,num} - T_{i,ana}\right|^2 dS_i)} \tag{4.6}$$

where $T_{i,num}$ and $T_{i,ana}$ are the numerical solution and analytical solution at point i, respectively. dS_i represents the area surrounding the point *i* and *S* is the total area of the domain. For this case, six levels of refinement have been used (from level 0 to level 5). The upper and lower bound of the action indicator is set to 0.1 and 0.01. During the computation, the adaptation checking procedure is activated for every 1500 iterations, to determine whether to insert or delete points. After the solution is converged, the node distributions using different resolution levels are shown in Figure 4.12, and the numerical performance of the stencil

adaptive method is quantitatively studied in Table 4.1. It can be observed from Figure 4.12 that with increasing resolution level, more nodes are added into the regions where high gradient of the solution is detected and that the node distributions can clearly display the distribution of the solution gradients. At the same time, it can be seen from Table 4.1 that increasing the resolution level also improves the accuracy of numerical solution. The level 0 grid achieves a solution with error of 9.02×10^{-3} while the level 4 grid achieves 6.89×10^{-4}. As compared with the fixed grid method (the local MQ-DQ method is also used) on a 160×40 grid, our adaptive algorithm with four resolution levels (4817 points totally) requires much less time to achieve a solution of similar accuracy. The solution of equation (4.4) is shown in Figure 4.13.

4.4.2 Natural convective heat transfer in a concentric annulus between a square outer cylinder and a circular inner cylinder

Natural convective heat transfer in enclosures is of great importance due to its wide applications in engineering. In this section, we will apply our developed local MQ-DQ based stencil adaptive method to simulate the natural convective heat transfer in a concentric annulus between a square outer cylinder and a circular inner cylinder.

4.4.2.1 Governing equations and boundary conditions

A schematic view of a horizontal concentric annulus between a square outer

cylinder and a heated circular inner cylinder is shown in Figure 4.14. Heat is generated uniformly within the circular inner cylinder (constant heat source), which is placed concentrically within the cold square cylinder (constant heat sink). If the cylinders are long enough and the flow reaches an equilibrium state, it can be considered to be steady, laminar and two dimensional. The buoyancy force is the driving force for the flow.

Based on the Boussinesq approximation, the non-dimensional governing equations for the problem are written in the vorticity-stream function formulation as

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \omega \tag{4.7}$$

$$u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = \Pr\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right) - \Pr Ra\frac{\partial T}{\partial x} \tag{4.8}$$

$$u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \tag{4.9}$$

where ψ denotes the stream function, ω represents the vorticity, and T is the non-dimensional temperature. The Prandtl number is defined as $\Pr = \frac{\mu C_p}{k}$, and the Rayleigh number is defined as $Ra = \frac{C_p \rho_0 g \beta L^3 \Delta T}{k v}$. Here, μ is the viscosity, C_p is the specific heat at constant pressure, k is the thermal conductivity, ρ_0 is the reference density, g is the gravitational acceleration, β is the thermal expansion coefficient, L is the side length of the square outer cylinder, ΔT is the temperature difference between the inner and outer cylinders, and ν is the

kinematic viscosity. Velocity components u and v can be computed from the stream function ψ as

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x} \tag{4.10}$$

Using the expressions in Equation (4.10), Equation (4.7) can also be written as

$$\omega = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \tag{4.11}$$

The governing equations (4.7)-(4.10) are discretized by the local MQ-DQ method. The discretization form of the governing equations at a general node *i* can be written as follows:

$$\sum_{k=1}^{N} w_{i,k}^{(2)} \psi_i^k + \sum_{k=1}^{N} \overline{w}_{i,k}^{(2)} \psi_i^k = \omega_i \tag{4.12}$$

$$u_i \sum_{k=1}^{N} w_{i,k}^{(1)} \omega_i^k + v_i \sum_{k=1}^{N} \overline{w}_{i,k}^{(1)} \omega_i^k = \Pr\left(\sum_{k=1}^{N} w_{i,k}^{(2)} \omega_i^k + \sum_{k=1}^{N} \overline{w}_{i,k}^{(2)} \omega_i^k\right) - \Pr Ra \sum_{k=1}^{N} w_{i,k}^{(1)} T_i^k \tag{4.13}$$

$$u_i \sum_{k=1}^{N} w_{i,k}^{(1)} T_i^k + v_i \sum_{k=1}^{N} \overline{w}_{i,k}^{(1)} T_i^k = \sum_{k=1}^{N} w_{i,k}^{(2)} T_i^k + \sum_{k=1}^{N} \overline{w}_{i,k}^{(2)} T_i^k \tag{4.14}$$

$$u_i = \sum_{k=1}^{N} \overline{w}_{i,k}^{(1)} \psi_i^k, \quad v_i = -\sum_{k=1}^{N} w_{i,k}^{(1)} \psi_i^k \tag{4.15}$$

where *k* is the *k*th supporting point of reference node *i*. As for the boundary conditions, the no-slip conditions are imposed on the wall, and both cylinders are considered isothermal. From the no-slip condition, the velocities u and v on both the inner and outer cylinder walls are zero. Thus the boundary condition can be written as follows:

$$u\big|_{inner\_wall} = u\big|_{outer\_wall} = 0, \quad v\big|_{inner\_wall} = v\big|_{outer\_wall} = 0 \tag{4.16}$$

$$\psi\big|_{inner\_wall} = 0, \quad \psi\big|_{outer\_wall} = 0 \tag{4.17}$$

$$T\big|_{inner\_wall} = 1\,,\quad T\big|_{outer\_wall} = 0 \tag{4.18}$$

The boundary condition for vorticity ω can be derived from Equation (4.11),

$$\omega\big|_{wall} = \left.\frac{\partial u}{\partial y}\right|_{wall} - \left.\frac{\partial v}{\partial x}\right|_{wall} \tag{4.19}$$

Like the governing equations, this boundary condition can also be discretized by the local MQ-DQ method and the vorticity value on the boundary can be updated by the following equation:

$$\omega_i = \sum_{k=1}^{N} \overline{w}_{i,k}^{(1)} u_i^k - \sum_{k=1}^{N} w_{i,k}^{(1)} v_i^k \tag{4.20}$$

4.4.2.2 Results and discussion

Definition of Nusselt numbers

The local heat transfer coefficient h is expressed as

$$h = -\mathrm{k}\frac{\partial \mathrm{T}}{\partial \mathrm{n}} \tag{4.21}$$

where k is the thermal conductivity. The average heat transfer coefficient $\overline{h}$ can be computed as

$$\overline{h} = \frac{1}{2\pi}\int_0^{2\pi} h\,d\xi \tag{4.22}$$

The average Nusselt numbers for the inner and the outer boundaries are, respectively, determined by

$$\overline{Nu_i} = \frac{\overline{h_i} S_i}{k}\,,\quad \overline{Nu_o} = \frac{\overline{h_o} S_o}{k} \tag{4.23}$$

where S_i and S_o are defined in the same way as in the work of Moukalled and Acharya (1996). In their work, the computational domain is taken as half of the physical domain due to symmetry, so S_i and S_o are taken as half of the

circumferential lengths of the inner and outer cylinder surfaces, respectively. As at steady state, the Nusselt numbers along the inner and outer walls are the same, there is no need to pay separate attention to $\overline{Nu_i}$ and $\overline{Nu_o}$. Thus in this study, we only show the value of $\overline{Nu_i}$, which is also noted as $\overline{Nu}$.

Validation of numerical results

In the present study, the Rayleigh number is fixed at 10^5 in a steady laminar flow regime, and the Prandtl number is set to be 0.71. For the natural convection problem, the temperature T is considered as an important flow parameter. Therefore, the temperature is chosen as the parameter of interest in this case. The upper and lower thresholds are set to 0.5 and 0.1, and the finest adaptive level is set to 3. During the computation, the adaptation checking procedure is activated for every 1000 iterations to determine whether to insert or delete points. All the differential operators are discretized by the local MQ-DQ method and the resultant algebraic equations are solved by the successive over-relaxation (SOR) iterative method.

In the literature, not much work has been done on the simulation of natural convection in a concentric annulus between a square outer cylinder and a circular inner cylinder. In the work of Moukalled and Acharya (1996), three different aspect ratios and four different Rayleigh numbers have been considered. Their numerical data were validated by comparison with some experimental data and

good agreement was found. In the present study, the results of Moukalled and Acharya (1996) will be used to validate the present numerical results. The maximum stream function value $\varphi_{\max}$ and the average Nusselt number $\overline{Nu}$ between the present work and the work of Moukalled and Acharya (1996) are compared in Table 4.2 for an aspect ratio of 2.5. It should be noted that due to different ways of non-dimensionalization between the work of Moukalled and Acharya (1996) and the present work, the equivalent $\varphi_{\max}$ in Table 4.2 is the one given from Moukalled and Acharya (1996) multiplied by the Prandtl number. From Table 4.2, we can see that the results with an initial grid of 80×36 show significantly larger values for all the three parameters than those by Moukalled and Acharya (1996). However, when the points are refined to resolution level 1, the results agree much better with the reference results. It indicates that with the refinement of the grid, the accuracy of the solutions has been improved. With regard to efficiency, we can compare the results of two cases. One is using the resolution level 1 grid whose initial grid is set to 80×36 and the other is using the fixed grid 160×71. From Table 4.2, it can be seen that as compared with the fixed grid method on a 160×71 grid (11360 grid points), our adaptive algorithm with one resolution level (4614 grid points) requires much less grid points and running time to achieve a solution of similar accuracy. The node distributions of using different resolution levels are shown in Figure 4.15 for an aspect ratio of 2.6. The isotherms and streamlines are displayed in Figures 4.16 and 4.17. Comparing Figures 4.15 and 4.16, we can see that with increasing resolution level,

more and more points are inserted into the regions near the inner circular cylinder and the regions near the top boundary of the outer square cylinder, where the variation of temperature is relatively large. A clear correlation between the refined regions in Figure 4.15 and the temperature gradients in Figure 4.16 can be observed.

4.5 Conclusions

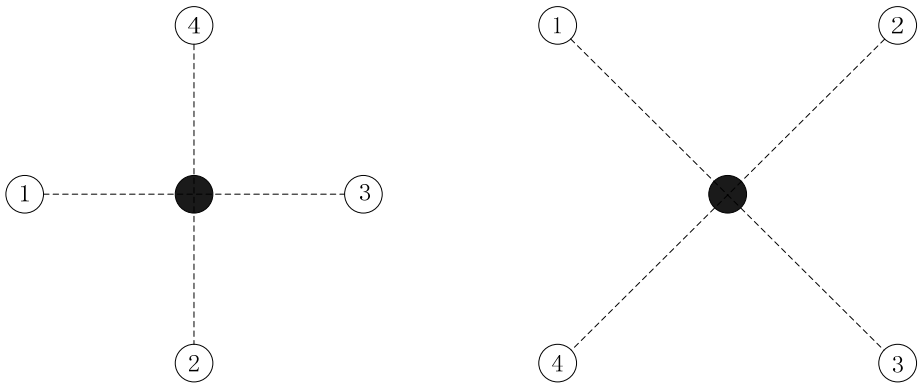
In this chapter, an efficient local stencil adaptive algorithm has been presented for two-dimensional fluid flow problems with curved boundaries. This algorithm is able to automatically adjust the local stencils to reflect the gradient of the solution. The local MQ-DQ method has been used to discretize the governing equations because of its meshless property. Two numerical experiments have been carried out to examine the performance of this method. Numerical results show that this method can effectively solve problems with curved boundaries. Furthermore, it can solve problems as accurately as the local MQ-DQ method does on a regular grid, but with less grid points and running time. As a result, the local MQ-DQ based stencil adaptive method offers a promising approach to solve engineering problems with curved boundaries.

Table 4.1 Numerical results of the adaptive Poisson solver

Maximum resolution level	**Grid points**	**Error_L2**	**Running time (seconds)** IBM ThinkPad R40 Intel Pentium M 1.5GHz 512MB memory
0	$40\times10=400$	9.02×10^{-3}	0.6
1	744	4.33×10^{-3}	1.3
2	1428	1.44×10^{-3}	5.2
3	2650	1.09×10^{-3}	17.5
4	4817	6.89×10^{-4}	71.0
Fixed node method	$160\times40=6400$	6.14×10^{-4}	177.0

Table 4.2 Comparison of $\varphi_{\max}$ and $\overline{Nu}$ for Aspect ratio $=2.5$, $\mathrm{Ra}=10^5$, $\mathrm{Pr}=0.71$

Different choices	**Number of Points used**	$\varphi_{\max}$	$\varphi_{\min}$	$\overline{Nu}$	**Running time (seconds)** IBM ThinkPad R40 Intel Pentium M 1.5GHz 512MB memory
80×36	2880	8.481	-8.481	5.102	230
Level 1	4614	8.350	-8.351	5.016	464
160×71	11360	8.347	-8.347	5.016	4017
Moukalled and Acharya (1996)	N.A.	8.38	N.A.	5.080	N.A.



Stencil type A　　Stencil type B

Figure 4.1 Configuration of two types of stencils

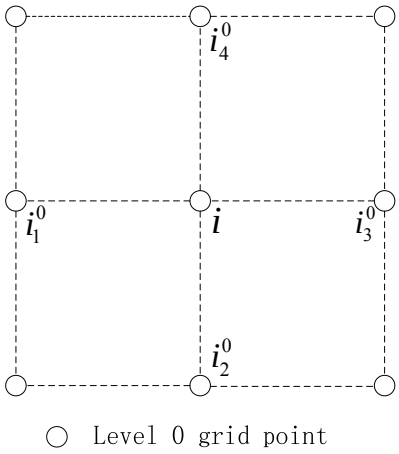


Figure 4.2 Configuration of an initial stencil

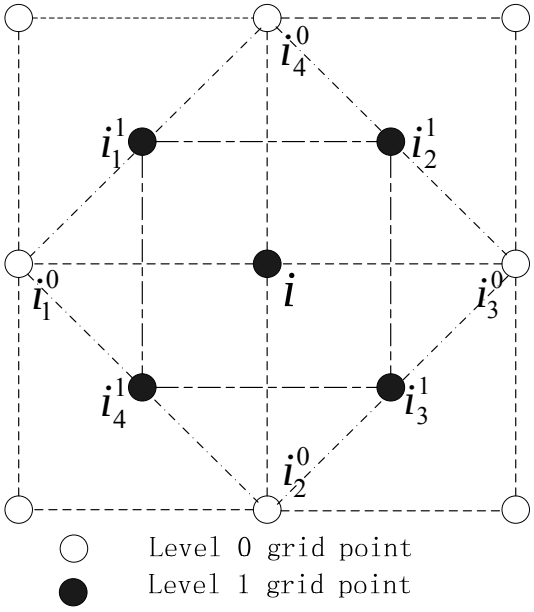


Figure 4.3 Stencil refinement from resolution level 0 to 1

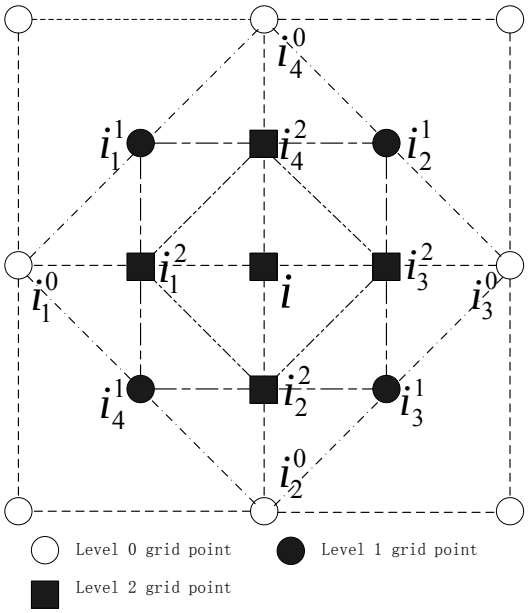


Figure 4.4 Stencil refinement from resolution level 1 to 2

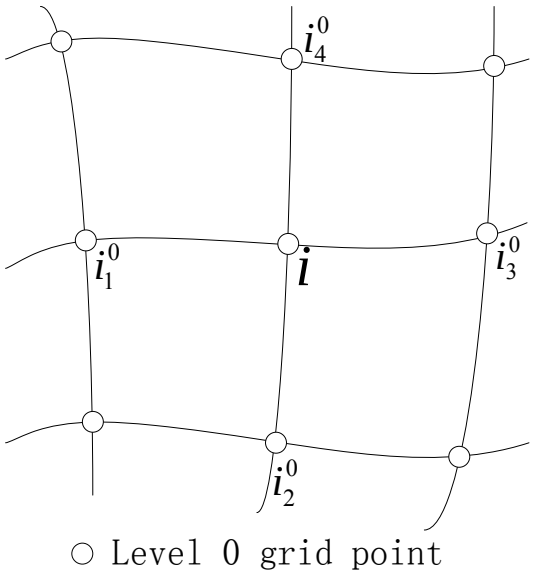


Figure 4.5 Configuration of an initial stencil in complex geometries

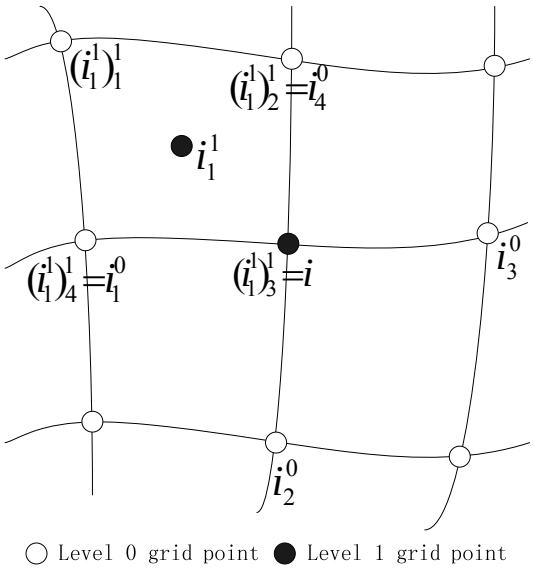


Figure 4.6 Injection of grid point i_1^1

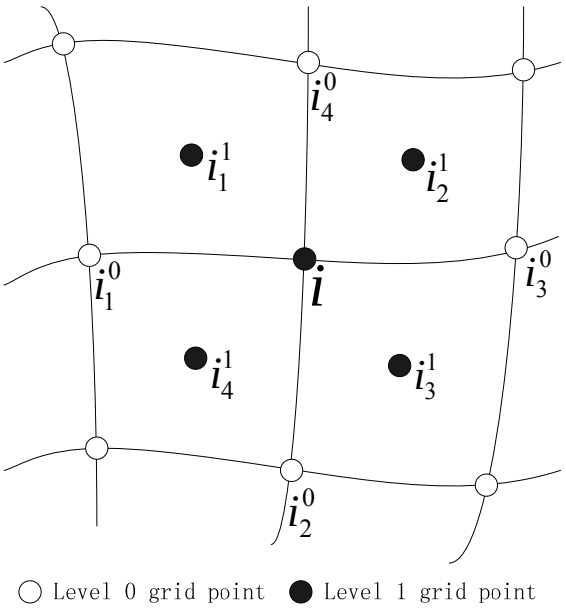


Figure 4.7 Stencil refinement from resolution level 0 to 1

i	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8

Figure 4.8 A one-dimensional array for storing supporting points

i	i_1^0	i_2^0	i_3^0	i_4^0	$(i_4^0)_1^0$	$(i_3^0)_4^0$	$(i_2^0)_3^0$	$(i_1^0)_2^0$

Figure 4.9 Initial eight supporting points for point i

i	i_1^1	i_2^1	i_3^1	i_4^1	i_1^0	i_2^0	i_3^0	i_4^0

Figure 4.10 Eight supporting points for point i in resolution level 1

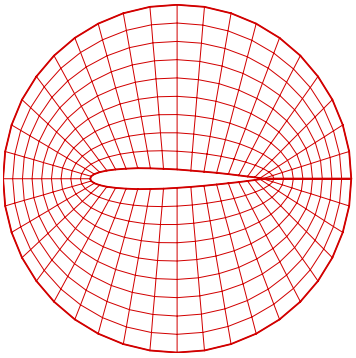
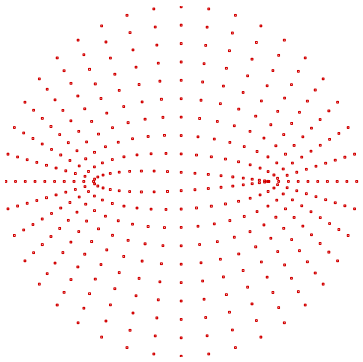
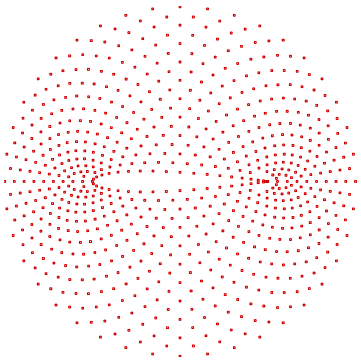


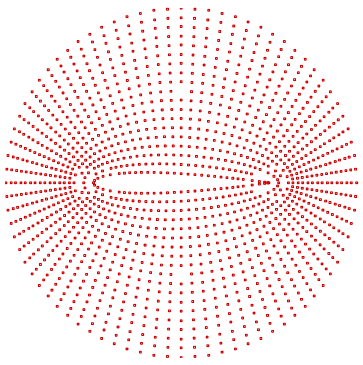
Figure 4.11 Domain around an airfoil and its background mesh



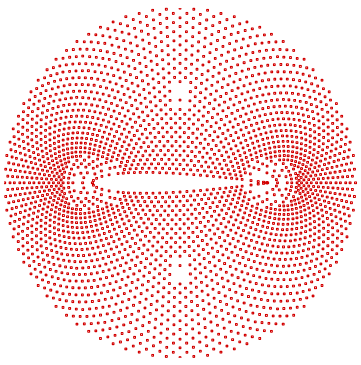
(a) $Level_{\max} = 0$



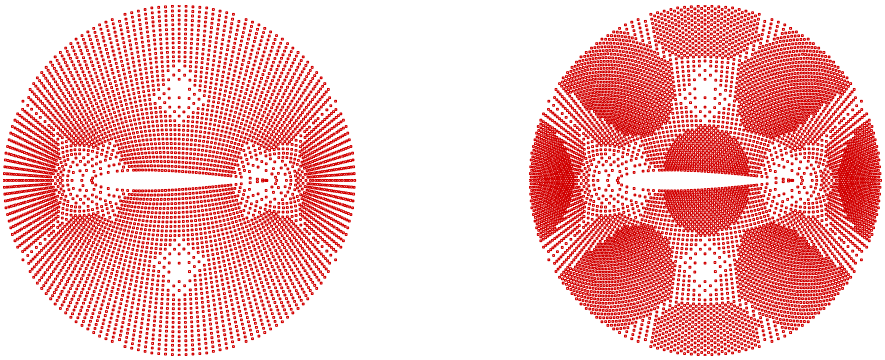
(b) $Level_{\max} = 1$



(c) $Level_{\max} = 2$



(d) $Level_{\max} = 3$



(e) $Level_{\max} = 4$ (f) $Level_{\max} = 5$

Figure 4.12 Final node distributions with different highest resolution levels

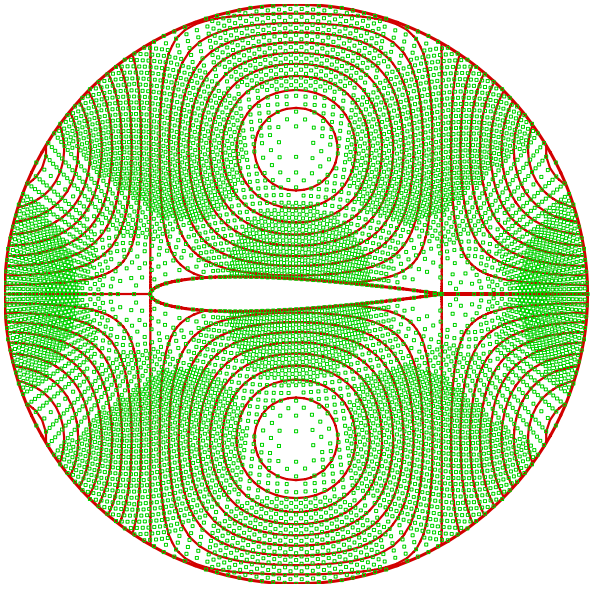


Figure 4.13 Contour of the variable T versus the node distribution with highest resolution level to 5, i.e., $Level_{\max} = 5$

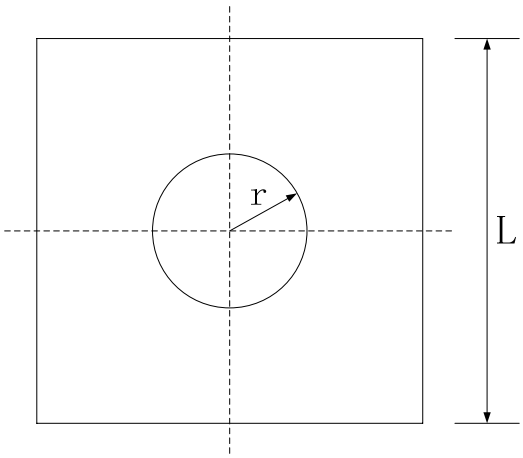
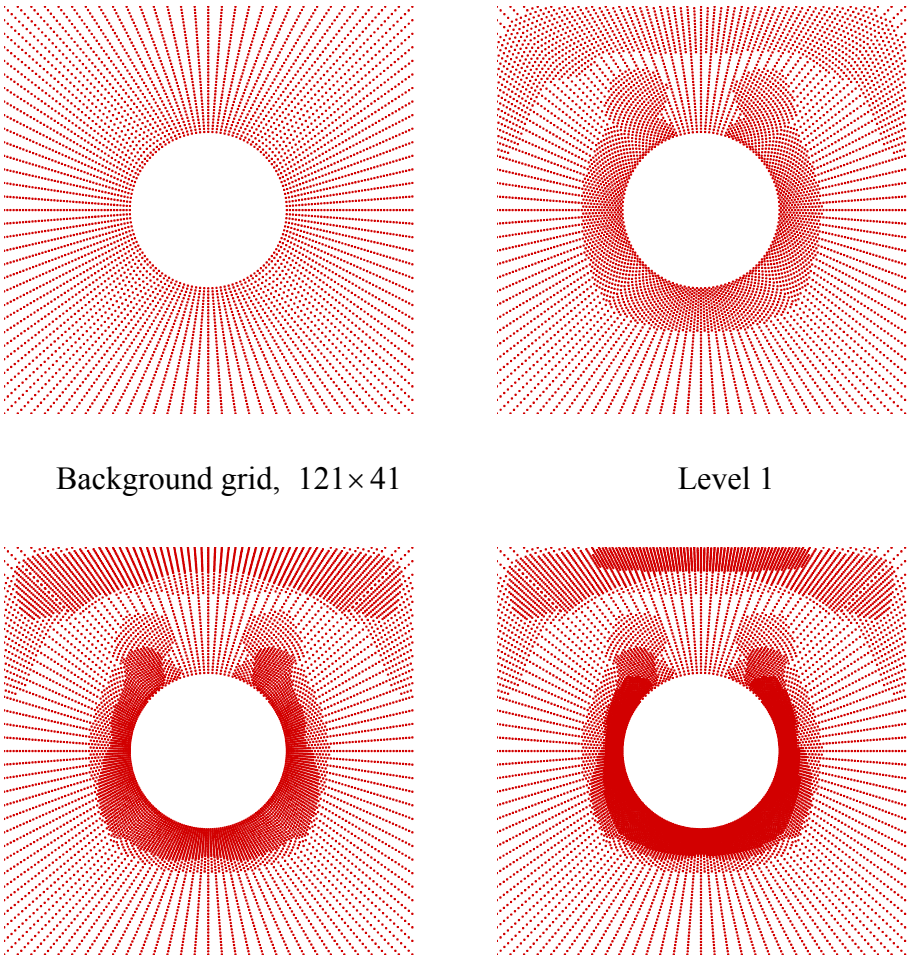


Figure 4.14 Sketch of physical domain of natural convection between a square outer cylinder and a circular inner cylinder



Background grid, 121×41 Level 1

Level 2 Level 3

Figure 4.15 Final node distributions with different highest resolution levels

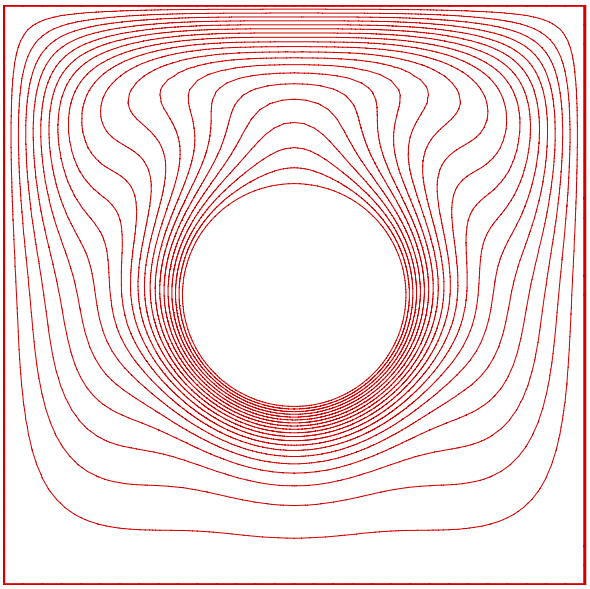


Figure 4.16 Isotherms for $\mathrm{Pr} = 0.71$, $Ra = 10^5$ and $rr = 2.6$

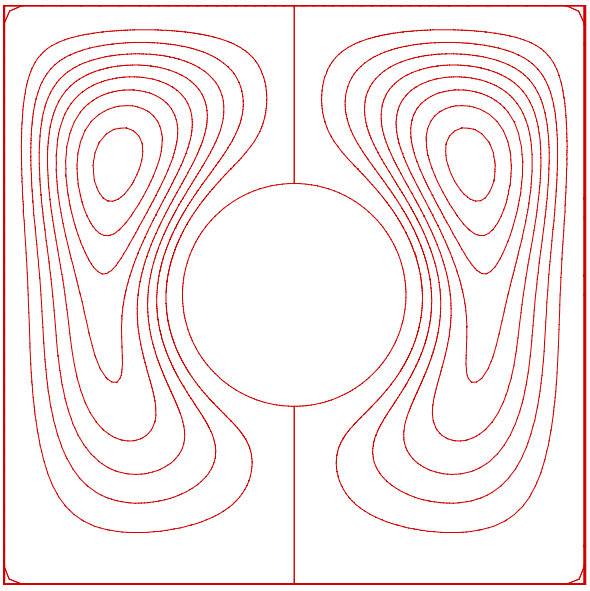


Figure 4.17 Streamlines for $\mathrm{Pr} = 0.71$, $Ra = 10^5$ and $rr = 2.6$

Chapter 5 Hybrid FD and Meshless Local MQ-DQ Method for Simulation of Viscous Flows around a Cylinder

In this chapter, an efficient hybrid method is presented for the simulation of two-dimensional steady and unsteady incompressible viscous flows. The method combines the conventional finite difference (FD) scheme and the meshless local multiquadric-differential quadrature (MQ-DQ) method. It takes the advantages of both methods, that is, the mesh-free property of the local MQ-DQ method and the high computational efficiency of the conventional FD scheme. In this method, the local MQ-DQ approach is applied for the spatial discretization in the region around the curved boundary. Correspondingly, FD scheme is applied in the rest of the flow domain. Numerical experiments are carried out for the simulation of flow past a circular cylinder with various Reynolds numbers. The obtained numerical results agree very well with computational and experimental data available in the literature. Numerical results showed that as compared with the fully local MQ-DQ method, the present method greatly improves the computational efficiency.

5.1 Motivation of this work

Although the local MQ-DQ method has gained much success in solving fluid flow problems, it does have some drawbacks which may hamper its further application. In general, it is less efficient than the conventional numerical methods such as finite difference (FD), finite element (FE) and finite volume (FV) methods. It was

shown by Shu et al. (2003) that in the local MQ-DQ method, any spatial derivative at a knot is approximated by a linear weighted sum of all the functional values in the local supporting region around the reference knot. As compared with the conventional finite difference methods, the local MQ-DQ method requires much more nodes to approximate a derivative with the same order of accuracy. This means that more computational efforts will be consumed to do simulations. This may be a main obstacle in industrial applications.

On the other hand, it is noted that although the conventional FD methods can approximate the derivatives very efficiently, they confront difficulties in dealing with complex geometries. Since FD methods can only be applied along a straight mesh line, when problems with complex geometry are considered, the so-called coordinate transformation technique should be implemented to map the physical domain into the computational domain. The mesh generation is very time-consuming, especially for problems with complex geometry in three-dimensional (3D) space. Furthermore, the original governing equations in the physical domain should also be transformed into relevant forms in the computational domain. The transformation procedure is usually very tedious and the transformed governing equations can be very complicated. In addition, further numerical errors will be brought into the discretization process due to the transformation procedure. Hence the accuracy of the solutions will be degraded as compared with the applications in regular domains. This is obviously the

bottleneck of the conventional FD methods in their industrial applications.

Taking into consideration the meshless property of the local MQ-DQ method and the high computational efficiency of FD methods, it seems that if these two methods are combined together, we can solve problems with complex geometry both effectively and efficiently. Based on this idea, we propose a method named hybrid FD and meshless local MQ-DQ method. In this method, the local MQ-DQ method is adopted for the spatial discretization in the region around the curved boundary and the conventional FD method is applied in the rest of the flow domain. Before implementation of this hybrid method, we can generate grid points very conveniently. For the area around the curved boundary, structured or unstructured grid points can be generated to fit the geometry of the solid body. Correspondingly, for the area with FD methods, Cartesian grid points can be easily generated. The grid generation techniques can be found in Ding et al. (2004), Perng and Street (1991) and Hinatsu and Ferziger (1991). To validate the present method, numerical simulation of steady and unsteady flows past a circular cylinder is considered. For this case, the cylindrical grid is used in a small region near the wall where the local MQ-DQ method is applied, while the Cartesian grid is used in the rest of the domain where the conventional FD schemes are applied. Numerical experiments show that this combination greatly improves the computational efficiency.

5.2 Hybrid FD and Meshless Local MQ-DQ Method

5.2.1 Local MQ-DQ method

The details of the local MQ-DQ method have been described in Chapter 2. In the local MQ-DQ method, the shape parameter c has a strong influence on the accuracy of numerical results. The optimal value of c is affected by the number of supporting points and the size of supporting region. Usually, the number of supporting points is fixed for an application. In this study, the number of supporting points is fixed as 16, which is considered to be able to give high order of accuracy from the experience of Shu et al. (2003). The size effect of supporting region on the choice of shape parameter c will be studied in Section 5.3.

5.2.2 Conventional FD scheme

Conventional FD scheme is the most popular numerical approach for discretization of derivatives. For the discretization of the first and second order derivatives which appear in the Navier-Stokes (N-S) equations for the fluid flow, the central difference scheme can be efficiently used on the uniform mesh. As we know, the usual procedure for deriving traditional FD schemes consists of approximating the derivatives in the differential equations via a truncated Taylor series expansion. In this study, the grid configuration for conventional FD schemes is shown in Figure 5.1. Functional values near node 0 can be approximated by the functional value and its derivatives at node 0 using the Taylor series expansion,

$$T_1 = T_0 + \left.\frac{\partial T}{\partial x}\right|_0 (-\Delta x_1) + \left.\frac{\partial^2 T}{\partial x^2}\right|_0 \frac{(-\Delta x_1)^2}{2} + \left.\frac{\partial^3 T}{\partial x^3}\right|_0 \frac{(-\Delta x_1)^3}{6} + O(\Delta {x_1}^4) \tag{5.1}$$

$$T_3 = T_0 + \left.\frac{\partial T}{\partial x}\right|_0 \Delta x_2 + \left.\frac{\partial^2 T}{\partial x^2}\right|_0 \frac{\Delta {x_2}^2}{2} + \left.\frac{\partial^3 T}{\partial x^3}\right|_0 \frac{\Delta {x_2}^3}{6} + O(\Delta {x_2}^4) \tag{5.2}$$

By $\Delta {x_1}^2 \times (5.2) - \Delta {x_2}^2 \times (5.1)$, we can get

$$\left.\frac{\partial T}{\partial x}\right|_0 = \frac{\Delta {x_1}^2 T_3 - \Delta {x_2}^2 T_1 - (\Delta {x_1}^2 - \Delta {x_2}^2) T_0}{\Delta {x_1}^2 \Delta x_2 + \Delta x_1 \Delta {x_2}^2} + O(\Delta^2) \tag{5.3}$$

And by $\Delta x_1 \times (5.2) + \Delta x_2 \times (5.1)$, we have

$$\left.\frac{\partial^2 T}{\partial x^2}\right|_0 = \frac{\Delta x_2 T_1 + \Delta x_1 T_3 - (\Delta x_1 + \Delta x_2) T_0}{\frac{\Delta {x_1}^2 \Delta x_2}{2} + \frac{\Delta x_1 \Delta {x_2}^2}{2}} + O(\Delta) \tag{5.4}$$

Similarly, the derivatives with respect to *y* can be written as:

$$\left.\frac{\partial T}{\partial y}\right|_0 = \frac{\Delta {y_1}^2 T_4 - \Delta {y_2}^2 T_2 - (\Delta {y_1}^2 - \Delta {y_2}^2) T_0}{\Delta {y_1}^2 \Delta y_2 + \Delta y_1 \Delta {y_2}^2} + O(\Delta^2) \tag{5.5}$$

$$\left.\frac{\partial^2 T}{\partial y^2}\right|_0 = \frac{\Delta y_2 T_2 + \Delta y_1 T_4 - (\Delta y_1 + \Delta y_2) T_0}{\frac{\Delta {y_1}^2 \Delta y_2}{2} + \frac{\Delta y_1 \Delta {y_2}^2}{2}} + O(\Delta) \tag{5.6}$$

Up to now, the formulations of the first and second order derivatives by conventional FD schemes have been obtained. Note that for the non-uniform mesh, the approximation of the second order derivatives given by equations (5.4) and (5.6) only has first order accuracy. It has second order accuracy on a uniform mesh. To achieve second order accuracy on a non-uniform mesh, equations (5.4) and (5.6) must be modified to include one more neighboring point. The derivation process is the same as above. In the following part, we will combine the conventional FD scheme with the meshless local MQ-DQ method to develop a new hybrid method.

5.2.3 Hybrid FD and meshless local MQ-DQ method

As shown above, if we approximate a derivative by the local MQ-DQ method, 17 nodes are required (including the reference node itself). This implies high computational costs. Comparatively, for the conventional FD scheme, only three nodes are used to approximate the derivative with second order accuracy. In this sense, the local MQ-DQ method is much less efficient than the conventional FD scheme. On the other hand, the local MQ-DQ method can be directly applied to solve problems with curved boundary. This is not the case for the conventional FD scheme. It seems that if the merits of both methods are combined together, we can solve problems with curved boundary both easily and efficiently. In this combination, the local MQ-DQ method is adopted to discretize spatial derivatives around curved boundary and the conventional FD scheme is responsible for the discretization of spatial derivatives in the rest of the domain.

The key point for the success of this hybrid method is the distribution of grid points. Two typical grid distributions for this method are shown in Figures 5.2 and 5.3. From these figures, we can see that a Cartesian mesh is generated as the background mesh. For the domain around the circular cylinder, grid points are distributed along concentric circles to make the points orthogonal to the circular boundary. For the region from the inner circular domain near the cylinder to the Cartesian mesh, grid points are generated by the software “Gambit” to make the

point distribution smooth. With grid points generated, the above hybrid method can be applied easily by applying conventional FD scheme on Cartesian grid points and the local MQ-DQ method on other grid points. Comparing Figure 5.2 with Figure 5.3, we can see that Cartesian mesh is uniform in Figure 5.2 and non-uniform in Figure 5.3. Both have their own advantages. With uniform Cartesian mesh, the central difference scheme can approximate the second order derivatives with second order accuracy. Comparatively, with non-uniform Cartesian mesh, the second order finite difference scheme for the second order derivatives is more complicated than the central difference scheme. On the other hand, the non-uniform mesh can distribute more points to regions where solutions vary rapidly and less points to regions with smooth solution. This type of point distribution is ideal for simulation of vortices. Moreover, with non-uniform Cartesian mesh, the mesh-free region can be greatly reduced, which will further improve the computational efficiency.

5.3 Choice of Shape Parameter *c* in Local MQ-DQ Method

As we know, the performance of the local MQ-DQ method is strongly dependent on the choice of the shape parameter c. Many attempts [Ding et al. (2005), Carlson and Foley (1991), Fornberg and Wright (2004), Golberg et al. (1996) and Wang and Liu (2002)] have been made to study the influence and choice of c on MQ approximation. In this study, we will particularly focus on the choice of an optimal c in the local MQ-DQ method.

In the present study, we fix the number of supporting points for each reference point and study the dependence of numerical error on the shape parameter and mesh size. The two-dimensional Poisson equation in a unit square is taken as a model problem, which can be written as:

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}=f(x,y) \quad \text{in} \quad \Omega=\{(x,y)\,|\,0\le x,y\le 1\},\ T=g \quad \text{on} \quad \partial\Omega \tag{5.7}$$

where f and g are determined in such a manner that the exact solution T of the Poisson equation is the given one. To study the performance of the local MQ-DQ method in simulating two classical types of flow problems: periodic boundary value problems and general boundary value problems, we take $T_1=\sin(\pi x)\sin(\pi y)$ and $T_2=x^2+y^2$ as the solution functions. Here, $T_1=\sin(\pi x)\sin(\pi y)$ can represent the solution of the periodic boundary value problems and $T_2=x^2+y^2$ can stand for the solution of the general boundary value problems.

To study the effect of mesh size, four uniform meshes of 21×21, 41×41, 61×61 and 81×81 are used in our computation. For each mesh, a number of different values of shape parameter c are chosen. After discretizing the Poisson equation by the local MQ-DQ method, we get a set of linear algebraic equations. To solve the resultant equations, successive over-relaxation (SOR) iteration method is used. The convergence criterion is set to 10^{-8}, which is considered to be small enough for a converged solution. When the solution is converged, its

accuracy is measured by the following relative L_2 error norm:

$$\|\varepsilon\| = \frac{\sqrt{\sum_{i=1}^{N}(u_{num} - u_{exact})^2}}{\sqrt{\sum_{i=1}^{N}(u_{exact})^2}} \tag{5.8}$$

The obtained numerical results are shown in Figure 5.4. From the figure, we can see that when the value of shape parameter is fixed, the finer grid will give the smaller numerical error. On the other hand, when the mesh size is fixed, with the increase of the value of shape parameter, the relative error first decreases. However, when the shape parameter increases to a certain value, the local MQ-DQ method will become unstable. This is consistent with the previous observation. Moreover, for the local MQ-DQ method, the finer the grid is, the smaller the shape parameter can be used. Comparatively, the coarser the grid is, the larger the shape parameter can be adopted.

In practical applications, grid points may be distributed unevenly. From the above analysis, we know that large shape parameter could be used in the region with coarse point distribution and small shape parameter must be adopted in the region with fine point distribution. As shown by Shu et al. (2003), the normalized local MQ-DQ method is designed to meet this requirement. However, it may be inappropriate in some situations, especially when grid point distribution changes very rapidly. As we know, in the normalized local MQ-DQ method, c is taken as

$$c = \bar{c} D_i \tag{5.9}$$

where c is the actual shape parameter, $\bar{c}$ is the normalized shape parameter

(given as a constant) and D_i represents the scale of the supporting region for point *i*. As $\bar{c}$ is fixed, c changes with the value of D_i. In the region with coarse point distribution, c could be large due to the large value of D_i. Comparatively, in region with fine point distribution, c could be small due to the small value of D_i. Furthermore, the value of c is proportional to each other with the proportional ratio of $\frac{D_i}{D_j}$, where D_j represents the scale of the supporting region for point *j*. This may cause some problems. Take the simulation of flow past a circular cylinder as an example. In this case, more grid points are distributed near the cylinder surface, where the density of grid points is very large. Comparatively, the density of grid points is very small in the region far away from the cylinder surface. If a small $\bar{c}$ is used, the value of c is not large enough to accurately simulate the flow around the circular cylinder. If a large $\bar{c}$ is used, however, the value of c is too large to simulate the flow far away from the cylinder. In other words, the computation will diverge. This is a challenging issue in using the normalized local MQ-DQ method to accurately simulate the flow problem.

To overcome this drawback, the normalized local MQ-DQ method will be replaced by the non-normalized local MQ-DQ method in this study. In this method, how to choose an appropriate c for each reference point is of most importance. According to our experience, we may use the same value of c_0 for each reference point in the computational domain at the very beginning. We may

try to choose the value of c_0 as large as possible before the computation becomes unstable. Usually, the value of c_0 is still not large enough in the region with coarse point distribution according to the above discussion. In this work, a new formulation

$$c = c_0 + w(\frac{D_i}{D_0} - 1) \tag{5.10}$$

is proposed to compute the shape parameter for each reference point, where D_0 represents the scale of the supporting region for the point with the smallest supporting region in the domain and w is the relaxation factor given by the end user. If $w = \mathrm{c}_0$, this method recovers its normalized counterpart. As compared with its normalized counterpart, this method is more flexible because we can change the value of w to make the simulation as accurate as possible without encountering numerical instability, thus making the local MQ-DQ method more stable.

5.4 Simulation of Steady and Unsteady Flows past a Circular Cylinder

In this section, the proposed hybrid method is used to simulate the laminar flow past a circular cylinder at different Reynolds numbers. It is well-known that for this problem, the flow maintains a stable pattern with a pair of symmetric counter-rotating vortices behind the cylinder at small Reynolds numbers, i.e., from 0 up to approximately 49. At moderate Reynolds numbers, i.e., from 50 up to approximately 190, though the flow remains laminar and two-dimensional, vortex shedding, also known as the Karman vortex street, can be observed. Because the

Karman vortex street developing behind the cylinder displays clearly time periodicity, this flow problem was often used as a prototype of unsteady separated flows. In this work, the Reynolds number ($\mathrm{Re} = U_\infty D/\upsilon$), based on the upstream velocity U_∞ and the diameter of the cylinder D, is selected to be 10, 20 and 40 for steady flows and 100, 200 for unsteady flows. The comparison of computational effort between the present approach and the fully local MQ-DQ method is also carried out in this section.

5.4.1 Governing equations and boundary conditions

The flow domain and boundary conditions are depicted in Figure 5.5. The non-dimensional governing equations, expressed in terms of vorticity ω and stream function ψ, are written as

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \omega \tag{5.11}$$

$$\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = \frac{1}{Re}\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right) \tag{5.12}$$

where *u*, *v* denote the components of velocity in the *x* and *y* direction, which can be calculated from the stream function

$$u = \frac{\partial \psi}{\partial y},\ v = -\frac{\partial \psi}{\partial x} \tag{5.13}$$

In this study, the in-flow velocity is specified with a free stream velocity U_∞, which is equivalent to imposing the boundary condition for stream function with

$\psi = U_\infty \cdot y$ and vorticity with $\omega = 0$. The top and bottom boundaries are located at a transverse distance of 16 times the cylinder diameter, which is assumed far enough to be the far-field according to the work of Behr et al. (1995). The boundary conditions imposed at these places are the same as the in-flow boundary. The out-flow boundary is located at a distance of 30 times of the cylinder diameter downstream of the rear of the cylinder. Neumann boundary conditions are imposed at the out-flow boundary.

As we know, in the multiply-connected domain, the value of stream function at the surface of stationary body is an unknown constant, which may vary with time. To determine this unknown constant, the single value pressure condition should be implemented according to the work of Tezduyar et al. (1988). After derivation, Ding et al. (2004) found that the value of stream function at the surface can be determined by the following equation

$$\int_0^{2\pi} (R\frac{\partial^3 \psi}{\partial r^3} + \frac{\partial^2 \psi}{\partial r^2}) d\theta = 0 \qquad (5.14)$$

The derivatives in equation (5.14) can be discretized by one-sided finite difference schemes and expressed in terms of the stream function at the cylinder wall and interior points. In this case, substituting the derivative approximation forms into equation (5.14), we can obtain the formulation of ψ_{wall} as

$$\psi_{wall} = -\frac{(\frac{3R}{2h} - \frac{1}{2})\sum_{i=1}^{N}\psi_{wall+2,i} + (4 - \frac{6R}{h})\sum_{i=1}^{N}\psi_{wall+1,i}}{N(\frac{9R}{2h} - \frac{7}{2})} \tag{5.15}$$

where N represents the number of points on the surface of the cylinder wall. R is the radius of the circular cylinder and h is the radius difference between cylindrical grid points, as depicted in Figure 5.6. Using equation (5.15), the stream function on the cylinder wall can be updated by the stream function values at the interior points, which are computed from the governing equations. With regard to the vorticity value on the cylinder wall, it can also be derived from one-sided finite difference scheme and written as

$$\omega_{wall,i} = \frac{3(\psi_{wall+1,i} - \psi_{wall,i})}{h^2} - \frac{1}{2}\omega_{wall+1,i} + O(h^2) \tag{5.16}$$

5.4.2 Definition of lift and drag coefficients

In a viscous flow, the lift and drag forces are determined from the pressure distribution and friction on the surface of the blunt body. For the cases of flow past a circular cylinder, the drag and lift can be obtained from

$$F_L = R\int(\mu R\frac{\partial\omega}{\partial n} - \mu\omega)\cos\theta d\theta \tag{5.17}$$

$$F_D = R\int(\mu R\frac{\partial\omega}{\partial n} - \mu\omega)\sin\theta d\theta \tag{5.18}$$

where R is the radius of the circular cylinder.

The non-dimensional drag and lift coefficients of the body are then given by

$$C_L = \frac{F_L}{\rho U_\infty^2 R},\quad C_D = \frac{F_D}{\rho U_\infty^2 R} \tag{5.19}$$

5.4.3 Efficiency comparison between present method and the fully local MQ-DQ method

In this part, we will study the improvement of the computational efficiency by the present method as compared with the fully local MQ-DQ method. Numerical experiment was carried out for Reynolds number of 20. Node distribution of numerical computation is shown in Figure 5.2. A four-stage Runge-Kutta method is adopted for temporal discretization. Table 5.1 presents the computational cost required to reach the steady-state solution by present hybrid method and the fully local MQ-DQ method. For all the three cases, the mesh-free region holds about 10.66% area of the whole domain and the distributions of Cartesian mesh points are the same. The only difference among the three cases is the density of grid points in the mesh-free region. It can be seen in Table 5.1 that when about 50% points are discretized by the conventional FD scheme, present hybrid method only uses 60% of the running time required for the fully local MQ-DQ method. This fact indicates that the present method leads to significant reduction in computational cost, and achieves significant improvement on the efficiency as compared with fully local MQ-DQ method. It is known that the efficiency improvement is mainly due to the less summation and multiplication required by the central difference scheme within one time step. Therefore, the extent of efficiency improvement mainly depends on the increase of percentage of points discretized by the conventional FD scheme. It can be observed in Table 5.1 that

the higher the percentage of the points to apply the conventional FD scheme, the greater the efficiency improved.

To decrease the percentage of the points used by the local MQ-DQ method, it is not feasible to keep on lowering the density of the grid points in the mesh-free region. It was found that when the mesh-free region is reduced, the percentage of the mesh-free points will be decreased and the efficiency of present method will be improved as compared with the local MQ-DQ method. However, when the mesh-free region is very small, the transition region between the mesh-free region and the FD region will get closer to the circular cylinder. As a consequence, the density of the grid points used by the FD scheme will be much larger. Thus, the total number of grid points may be very large and the efficiency of the method would be impaired. Based on this consideration, non-uniform Cartesian grid points, which are shown in Figure 5.3, may be generated as alternative grid distribution for the FD application. It can be seen in Figure 5.3 that the mesh-free region as well as the number of mesh-free points may be greatly reduced. Moreover, more points can be distributed at the rear of the circular cylinder where the so-called Karman vortex street develops and less points in the region with smooth solutions. This type of grid distribution may help to solve the problem both effectively and efficiently.

5.4.4 Simulation of steady flow at low Reynolds numbers

Numerical simulations were carried out for three small Reynolds numbers: 10, 20 and 40, respectively. The streamlines when flow reaches its final steady state are illustrated in Figure 5.7. From these plots, we can see that a pair of vortices develops behind the cylinder and is perfectly aligned. This is consistent with previous observation. Furthermore, some quantitative parameters for the recirculating region, such as the length of the recirculating region L_{sep}, separation angle θ_{sep} and drag coefficient C_d are computed and listed in Table 5.2. The obtained results are compared with those from other researchers [Ding et al. (2004), Takami and Keller (1969), Dennis and Chang (1970), Tuann and Olson (1978)]. It can be seen from Table 5.2 that all these flow parameters agree well with the results of previous studies.

5.4.5 Simulation of unsteady flow at moderate Reynolds numbers

In this section, the hybrid method is applied to solve flow past a circular cylinder with Reynolds numbers of 100 and 200. In this test case, a total of 17454 grid points are used and the time step is set to 0.002. The value of shape parameter for each reference point is determined as $c = 0.2 + 0.1(\frac{D_i}{D_0} - 1)$. A number of comparisons are made between the present study and the work of Ding et al. (2004), Braza et al. (1986) and Liu et al. (1998).

For the flow past blunt bodies, the drag and lift coefficients on the surface of body are two very important parameters. The time-evolution of these two characteristic

parameters illustrate the variation of the flow field. It can be observed in Figures 5.8 and 5.9 that lift and drag coefficients show obvious periodic oscillations for both $\mathrm{Re} = 100$ and 200 cases. This implies the periodic variation of flow field. From Figures 5.8 and 5.9, it can also be found that the lift coefficient oscillates with larger amplitude than the drag coefficient, and the drag coefficient varies twice as fast as the lift coefficient. These phenomena are consistent with those observed by other researchers. It is because the drag coefficient is affected by vortex shedding process from both sides of the cylinder. In addition, the present results are quantitatively compared with the numerical results by other researchers in Tables 5.3 and 5.4. It can be observed that good agreement has been achieved.

The Strouhal number (St) is another important parameter which needs to be considered. St is used as a measure of the oscillating fluid flow phenomenon in the wake region. It is connected to the flow parameter by the following relationship

$$St = f_s \cdot D / U_\infty \tag{5.20}$$

where f_s is the shedding frequency. Since D and U_∞ are non-dimensionalized as unity, S_t is equivalent to the shedding frequency. Thus, the Strouhal number is obtained as the inverse of the time period of vortex shedding. In the present study, Strouhal numbers are deduced from the time-evolution of the lift coefficient C_l. It can be seen from Tables 5.3 and 5.4 that, the S_t number computed using our algorithm is very close to other researchers' result. Figures 5.10 and 5.11 present the streamlines and vorticity contours for unsteady cases of $\mathrm{Re} = 100$ and 200,

respectively. The streamlines indicate the flow pattern at a given instant of time, and the vorticity contours illustrate the instantaneous vortex structure in the wake region.

5.5 Concluding Remarks

In this chapter, a hybrid FD and meshless local MQ-DQ method was proposed to solve two-dimensional incompressible viscous flows. This method combines the advantages of the finite difference method and the local MQ-DQ method, and can thus be applied to solve problems with curved boundary easily and efficiently. It is validated by simulating the steady and unsteady flow past a circular cylinder. The obtained results were in line with other experimental and numerical studies. In addition, the results show that the hybrid method greatly improves the computational efficiency as compared with the fully local MQ-DQ method.

The effect of shape parameter c and mesh size on the performance of the local MQ-DQ method was also studied in this chapter. We found that large shape parameter can be used in the region with coarse point distribution and small shape parameter should be used in the region with fine point distribution. Based on this observation, an empirical expression of $c = c_0 + w(D_i / D_0 - 1)$ was proposed to determine the shape parameter for each reference point. With this empirical expression, the local MQ-DQ method was found to become more stable.

Table 5.1 Efficiency comparison between local MQ-DQ method and present method for Re = 20

Total node number	Method	Percent of FD points	CPU-time (s)	Efficiency improvement
12999	Present	32.5	1196	25.81%
	MQ-DQ	0	1612	
10568	Present	40.1	880	32.57%
	MQ-DQ	0	1305	
8989	Present	47.4	672	39.13%
	MQ-DQ	0	1104	

Table 5.2 Comparison of length of the recirculating region (L_{sep}), separation angle (θ_{sep}) and drag coefficient (C_d) for *Re* = 10, 20 and 40

Re	Source	L_{sep}	θ_{sep}	C_d
10	Takami and Keller (1969)	0.249	29.3	2.80
	Dennis and Chang (1970)	0.252	29.6	2.85
	Tuann and Olson (1978)	0.25	29.7	3.18
	Ding et al. (2004)	0.252	30.0	3.07
	Present	0.248	29.7	2.92
20	Takami and Keller (1969)	0.935	43.7	2.01
	Dennis and Chang (1970)	0.94	43.7	2.05
	Tuann and Olson (1978)	0.9	44.1	2.25
	Ding et al. (2004)	0.93	44.1	2.18
	Present	0.96	43.2	2.10
40	Takami and Keller (1969)	2.32	53.6	1.536
	Dennis and Chang (1970)	2.35	53.8	1.522
	Tuann and Olson (1978)	2.1	54.8	1.675
	Ding et al. (2004)	2.20	53.5	1.713
	Present	2.26	52.8	1.552

Table 5.3 Comparison of drag coefficient and lift coefficient for $Re = 100$

	Cd	*Cl*	*St*
Braza et al (1986)	1.364 ± 0.015	± 0.25	0.160
Liu et al. (1998)	1.350 ± 0.012	± 0.339	0.164
Ding et al. (2004)	1.325 ± 0.008	± 0.28	0.164
Present	1.355 ± 0.009	± 0.285	0.168

Table 5.4 Comparison of drag coefficient and lift coefficient for $Re = 200$

	Cd	*Cl*	*St*
Braza et al (1986)	1.40 ± 0.05	± 0.75	0.200
Liu et al. (1998)	1.31 ± 0.049	± 0.69	0.192
Ding et al (2004)	1.327 ± 0.045	± 0.60	0.196
Present	1.348 ± 0.047	± 0.65	0.197

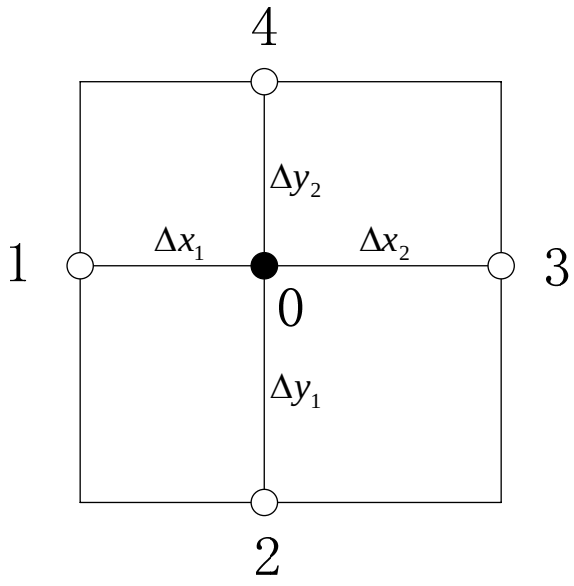
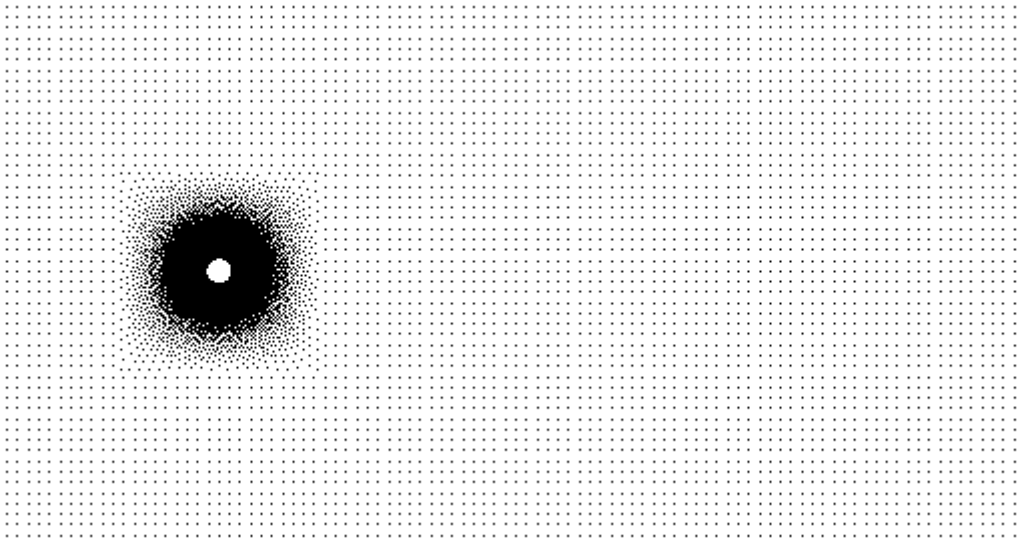
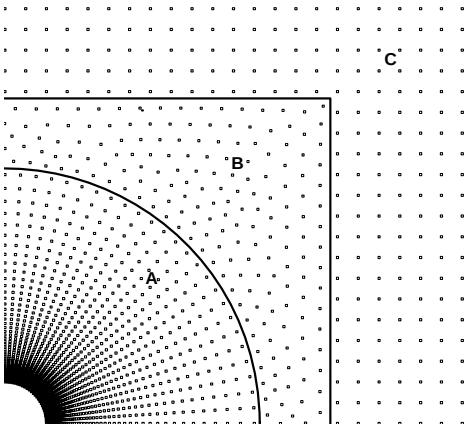


Figure 5.1 Grid configuration for conventional FD schemes



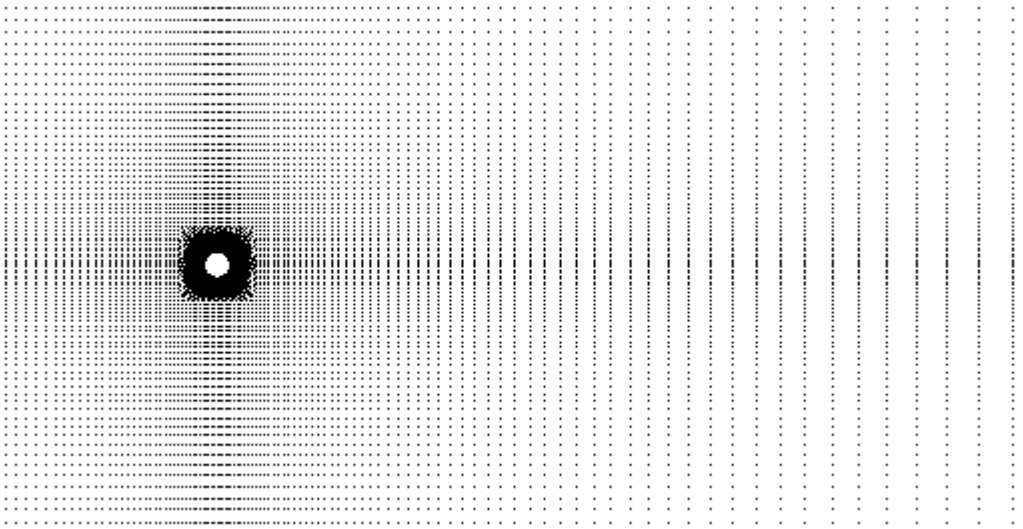
a. Grid distribution in the whole computational domain



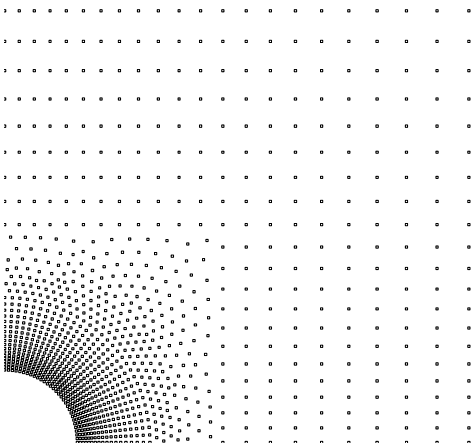
b. Grid distribution around the circular cylinder: A. Cylindrical grid points

B. Scattered grid points C. Cartesian grid points

Figure 5.2 Grid distribution with uniform Cartesian mesh points

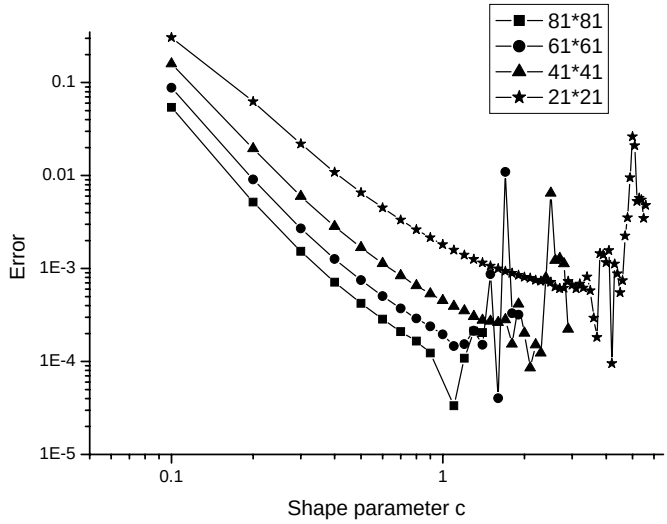


a. Grid distribution in the whole computational domain

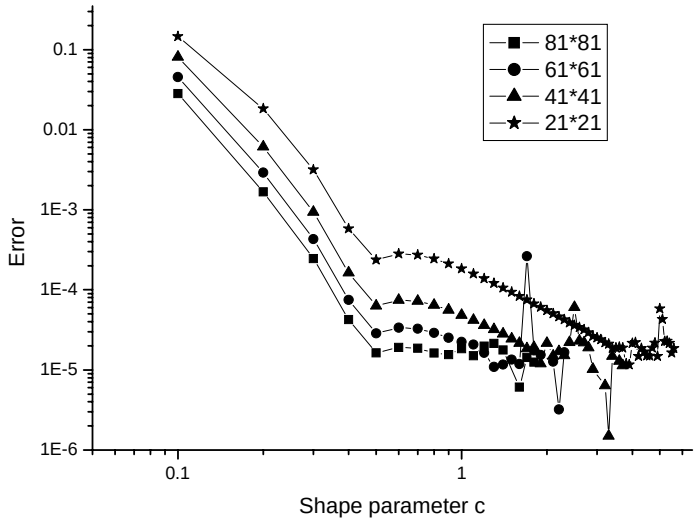


b. Grid distribution around the circular cylinder

Figure 5.3 Grid distribution with non-uniform Cartesian mesh points



(a) $\sin(\pi x)\sin(\pi y)$



(b) $x^2 + y^2$

Figure 5.4 Convergence history of relative error versus shape parameter c

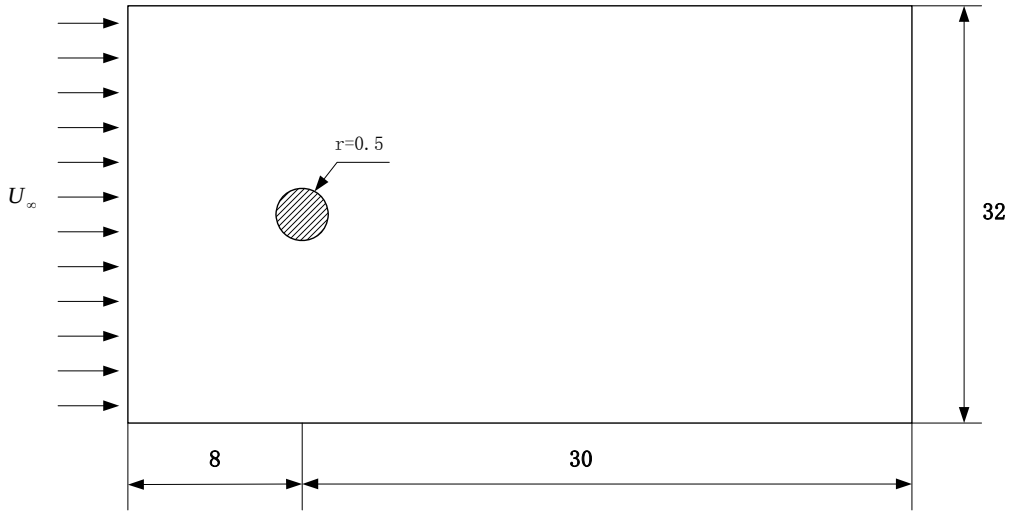
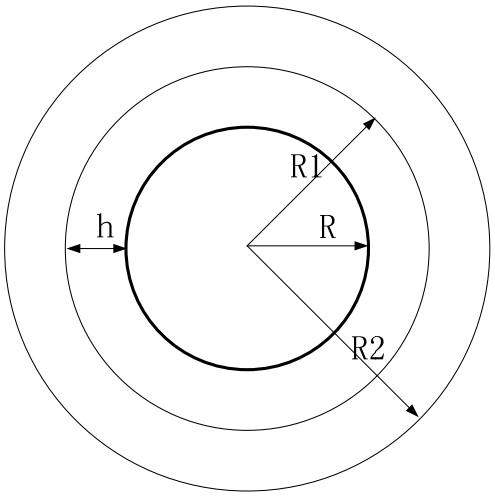
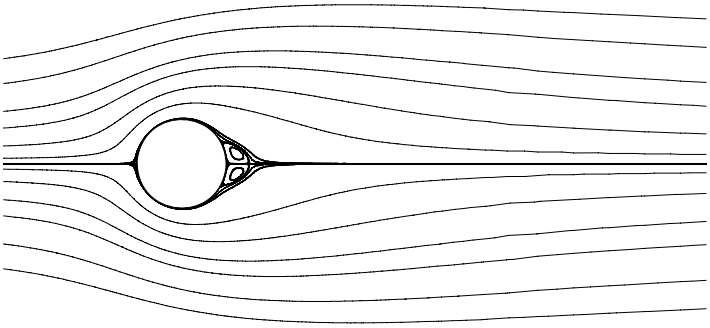


Figure 5.5 Configuration of flow around one isolated cylinder

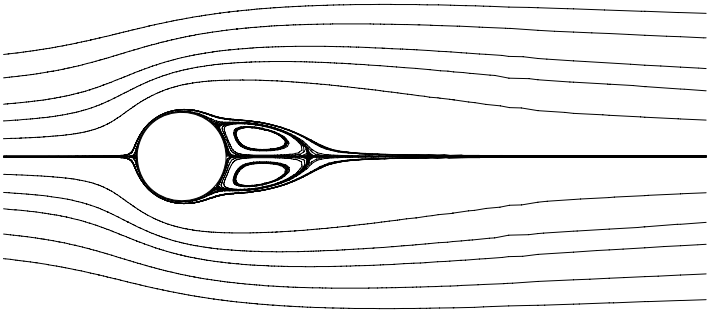


$$h = R2-R1 = R1-R$$

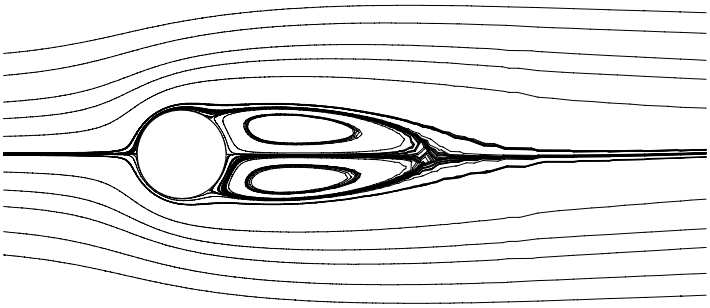
Figure 5.6 A local body-fitted coordinate system



Re = 10



Re = 20



Re = 40

Figure 5.7 Streamlines for *Re* = 10, 20 and 40

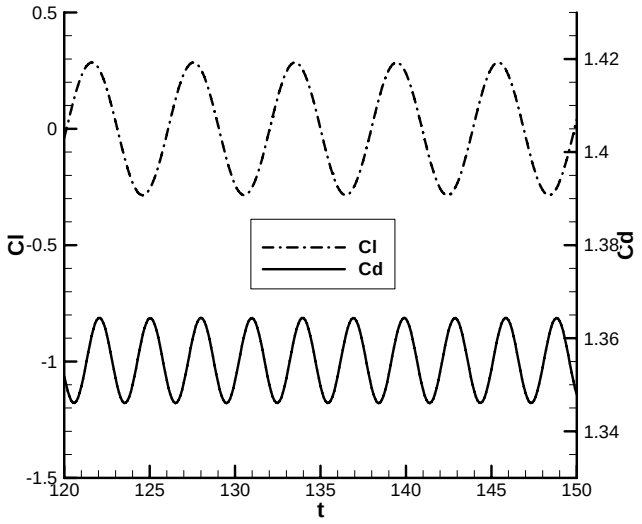


Figure 5.8 The time-evolution of lift and drag coefficients for $\mathrm{Re} = 100$

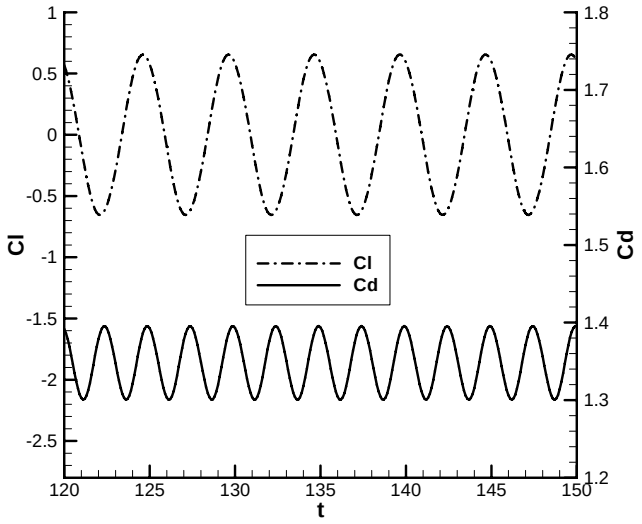


Figure 5.9 The time-evolution of lift and drag coefficients for $\mathrm{Re} = 200$

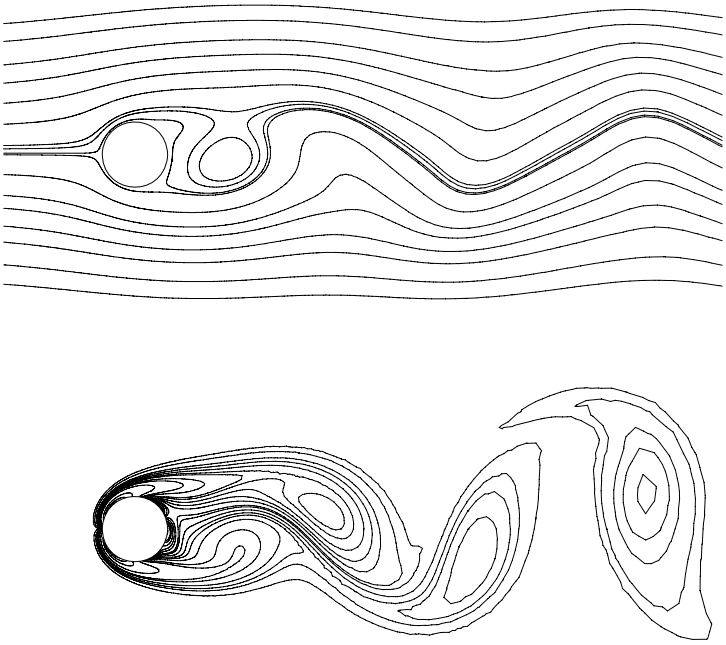


Figure 5.10 Streamlines and vorticity contours for $Re = 100$

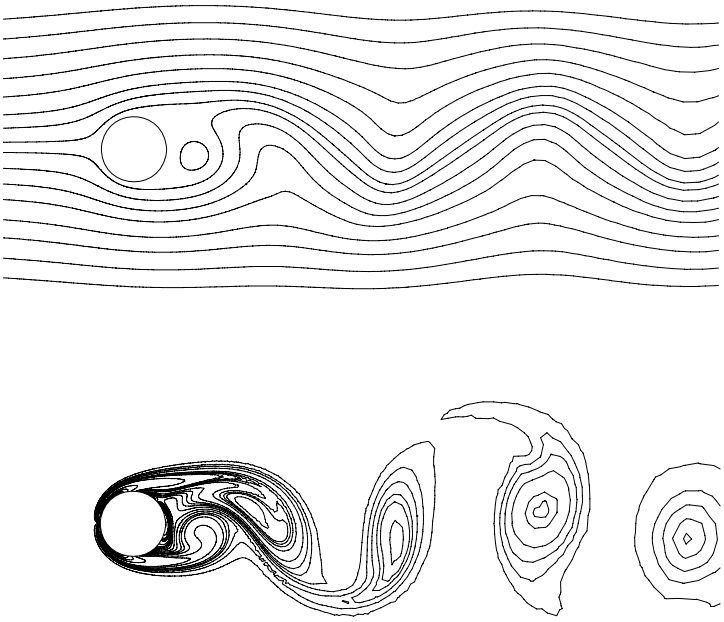


Figure 5.11 Streamlines and vorticity contours for $Re = 200$

Chapter 6 Application of Local MQ-DQ Method to Solve 3D Incompressible Viscous Flows with Curved Boundary

In this chapter, the local MQ-DQ method is extended to simulate fluid flow problems with curved boundary in three-dimensional (3D) space. The main concern of this work is to numerically study the performance of the 3D local MQ-DQ method and demonstrate its capability and flexibility for simulation of 3D incompressible fluid flows with curved boundary. Fractional step method is adopted for the solution of Navier-Stokes (N-S) equations in the primitive-variable form. Flow past a sphere at various Reynolds numbers is chosen as a test case to validate the 3D local MQ-DQ method. The computed solution is compared well with available data in the literature. The numerical solution shows that the local MQ-DQ method can be applied to solve incompressible viscous flow problems with curved boundary in 3D space effectively. Moreover, a lid-driven flow in a cubic cavity with a stationary, rigid sphere at its center is studied in detail by the local MQ-DQ method.

6.1 Motivation of this work

Despite the success of local MQ-DQ method in solving fluid flow problems, most of the work up to now only focused on two dimensional cases. Very little work has been done to explore its capability of solving three dimensional (3D) fluid flow problems, especially with curved boundary. Ding et al (2006) applied the

local MQ-DQ method to solve a classical driven cavity flow problem in 3D space and proved the feasibility of this method for 3D fluid flow problems. However, the case they studied is very simple and the geometry of the problem is quite regular (cubic). As we know, the main advantage of RBF-based schemes is their mesh-free property and their flexibility for handling problems with curved boundary. Adopting the local MQ-DQ method to solve the 3D driven cavity flow problem, however, cannot reflect its advantage.

In this study, we will extend the local MQ-DQ method to solve 3D fluid flow problems with curved boundary. At this time, the theoretical error analysis for derivative approximation by RBF is not available. Thus, firstly, we will make an error estimate for the derivative approximation by the 3D local MQ-DQ method to provide a useful guidance for implementation of this method. Actually, this work is the 3D counterpart of that by Ding et al (2005). Ding et al. (2005) found the influence of shape parameter and supporting points on numerical accuracy of the 2D local MQ-DQ method through numerical experiments. In this work, while applying the 3D local MQ-DQ method to solve fluid flow problems, we will select shape parameter and supporting points based on the 3D results of the error estimates.

The 3-D flow past a sphere will be solved by the local MQ-DQ method to demonstrate its capability and flexibility in solving 3D flow problems with curved boundary. This problem has been studied by a number of researchers, and can thus be selected as a model problem to test a new numerical method. As has been discussed in Chapter 5, the main drawback of the local MQ-DQ method is that it is less efficient than the conventional numerical methods such as finite difference (FD), finite element (FE) and finite volume (FV) methods. Compared with the simulation of 2-D cases, this drawback is intensified in solving 3-D problems. This method may be too costly in terms of computational efforts and memory requirements to be applicable in real industrial problems. Thus, the hybrid FD and meshless local MQ-DQ method proposed in Chapter 5 will be adopted in this 3-D simulation to reduce the computational cost. With this hybrid technique, the 3-D local MQ-DQ method can be applied to solve 3-D problems with curved boundary both flexibly and efficiently.

6.2 Error Estimates of the 3-D Local MQ-DQ Method

During the implementation of local MQ-DQ method, it is interesting to know the order of accuracy for the derivative approximation by the 3D local MQ-DQ method. Furthermore, the number of supporting points and free shape parameter c need to be determined by end-users. However, due to the lack of theoretical analysis, there is no error estimate available for this method to date. Thus, how to

determine the number of supporting points and free shape parameter *c* poses a challenging issue for the end-users.

In this section, we mainly focus on error estimates of the derivative approximation by the 3D local MQ-DQ method to explore the relationship between the approximation error, the number of supporting points and free shape parameter *c*. Based on the study, we can provide a useful guidance for the implementation of the 3D local MQ-DQ method. The performed analysis of error estimates is based on numerical experiments. In fact, this study is the extension of the work by Ding et al. (2005). Ding et al. (2005) performed error estimates of the second order derivative approximation by the local MQ-DQ method in 2D space through the solution of 2D Poisson equation. They found the relationship between the approximation error, the number of supporting points and free shape parameter *c*. In this work, numerical experiment is also made to find the relationship between the numerical error of the second order derivative approximation and the two factors.

3D Poisson equation which only has the second order derivatives is taken as a model problem, and can be written as:

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}+\frac{\partial^2 T}{\partial z^2}=f(x,y,z)\ \text{ in }\ \Omega=\{(x,y,z)\,|\,0\le x,y,z\le 1\}\,,\ T=g\ \text{ on }\ \partial\Omega \quad (6.1)$$

where f and g are determined in such a manner that the exact solution T of the Poisson equation is the given one. To study the performance of the 3D local MQ-DQ method in simulating two classical types of flow problems: periodic boundary value problems and general boundary value problems, we take $T_1 = \sin(\pi x)\sin(\pi y)\sin(\pi z)$ and $T_2 = x^4 + y^4 + z^4$ as two typical solution functions. Here, T_1 can represent the solution of periodic boundary value problems and T_2 can stand for the solution of general boundary value problems.

In order to estimate the approximation error, it is necessary to understand the relationships between the numerical error and the two factors. To fulfil this goal, numerical experiments are designed in such a manner that one factor is fixed and the other one is variable. In this work, we firstly study the relationship between the numerical error and the number of supporting points. This means that the free shape parameter c is fixed, and through varying the number of supporting points, we can obtain its relationship with the numerical error. Then, we study the relationship between the numerical error and the free shape parameter c. In this process, the number of supporting points is fixed. We get its relationship with numerical error through varying the value of free shape parameter c. The effect of boundary conditions on the accuracy of numerical solution is not considered in this work.

6.2.1 Relationship between numerical error and number of supporting points

For the investigation of relationship between the numerical error and the number of supporting points n_s, the value of free shape parameter *c* is fixed as 0.2, which is suitable for all the cases considered. For each n_s, 3D Poisson equation is solved on five different uniform meshes, i.e. $31\times31\times31$, $41\times41\times41$, $51\times51\times51$, $61\times61\times61$ and $71\times71\times71$. Thus the convergence rate of relative error versus the mesh size *h* for each n_s can be studied. The numerical solutions are illustrated in Figure 6.1 in the form of relative error versus *h* in the log-log scale.

It can be observed in Figure 6.1 that the symbols representing the accuracy of solution with the same n_s are in perfect alignment. This implies that the 3D local MQ-DQ method accomplishes the so-called super convergence, i.e. an error estimate of $O(h^n)$. Furthermore, from Figure 6.1, we can also see that the convergence lines can be classified into two groups by the value of slope, with the number of supporting points ranging from 6 to 36. Specifically, the convergence rate is approximately 2.0 for the scheme with less than 31 supporting points and 3.9 for the scheme with 32 and 36 supporting points. Therefore, an error estimate with respect to the mesh size *h* and the number of supporting points can be written as

$$\|\varepsilon\| \sim O(h^n) \text{ and } n \approx \begin{cases} 2.0 & \textit{for } 6 \le \mathbf{n_s} \le 31 \\ 3.9 & \textit{for } 32 \le \mathbf{n_s} \le 36 \end{cases} \tag{6.2}$$

The above results show that the accuracy of the second order derivative approximation can be greatly improved at some critical number of supporting points. The mean values of convergence rate for each n_s are listed in Table 6.1. From Table 6.1, it can be seen that the convergence rates are independent of solution functions. In our work, the maximum n_s tested is 36. This is because in practical applications, the n_s value is usually taken below 36.

6.2.2 Relationship between numerical error and free shape parameter *c*

To study the relationship between numerical error and the free shape parameter *c*, the number of supporting points is fixed at 32, based on the above observation. Five different values of free shape parameter *c* are tested, ranging from 0.1 to 0.3. For each *c*, 3D Poisson equation is solved on five different uniform meshes. The numerical solutions are shown in Figures 6.2(a)-(b) for the two solution functions, respectively. In each figure, the numerical results are plotted in the form of relative error versus mesh size in the log-log scale and a group of convergence lines are drawn according to the same value of shape parameter *c*. The mean values of convergence rate of relative error versus the mesh size *h* are listed in Table 6.2. From Figures 6.2(a)-(b), it is clear to see that the convergence lines standing for different shape parameters are parallel to each other. In other words, they have the same convergence rate (order of accuracy). This can also be confirmed by the mean value of convergence rate listed in Table 6.2. Therefore,

from the viewpoint of convergence rate, we can say that the contributions of shape parameter *c* and mesh size *h* are utterly independent. From Table 6.2, it can be seen that the convergence rates for the different solution functions with various shape parameter *c* almost have the same value (≈ 3.9). This confirms our previous finding for the convergence property with respect to mesh size *h*, i.e. the convergence rate with respect to mesh size *h* is independent of the solution function.

6.3 Numerical Procedure for Simulating Flows past a Sphere

To demonstrate the ability of local MQ-DQ method for simulation of a flow with curved boundary in 3D space, we apply it to simulate laminar flow past a sphere for $20 \le \mathrm{Re} \le 270$. The Reynolds number Re is based on the sphere diameter *D* and the free stream velocity *U*. The flow of a viscous fluid past a stationary isolated sphere may be considered as a simplified case of a general family of immersed bluff-body flows with widespread applications. The behaviour of the flow past a sphere at different Reynolds numbers has been studied by a number of researchers. In our study, we can easily compare our results with those in the literature to validate the 3D local MQ-DQ method.

6.3.1 Hybrid FD and local MQ-DQ method

As shown above, if we approximate a derivative by the local MQ-DQ method, 33 nodes are required (including the reference node itself) to get high accuracy. That implies high computational costs. Comparatively, for the conventional FD scheme, much less number of grid points is needed to get the same order of accuracy. In this sense, the local MQ-DQ method is less efficient than the conventional FD scheme. It seems that, if we only apply the local MQ-DQ method in the domain around the curved boundary and apply the conventional FD scheme in the remaining domain, we can solve problems with curved boundary both easily and efficiently.

To combine the conventional FD method and the local MQ-DQ method together, we have to make sure that the distribution of grid points is proper for this hybrid approach. In this case, the sphere is located at the origin of the Cartesian coordinate system in a rectangular box, $-5 \le x \le 10$, $-5 \le y \le 5$ and $-5 \le z \le 5$. The grid distribution is shown in Figure 6.3, in which 510768 nodes are used, and the minimum and maximum distances between two neighbouring nodes are respectively 0.025 and 0.125. For the convenience of illustration, only the grid points on the *x*-*y* plane at $z = 0$ are plotted. From this figure, it can be seen that a Cartesian mesh is generated as the background mesh. For the domain around the sphere, grid points are distributed along concentric spheres to make the points orthogonal to the solid boundary. As shown in Figure 6.3, for the region between

the inner sphere domain and outer domain, grid points are generated by the software “GAMBIT” to make the grid distribution smooth.

6.3.2 Governing equations

The three-dimensional, unsteady, incompressible Navier-Stokes equations in the primitive variable form are taken as the governing equations, which are written as

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}=-\frac{\partial p}{\partial x}+\frac{1}{\mathrm{Re}}(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}) \tag{6.3a}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}=-\frac{\partial p}{\partial y}+\frac{1}{\mathrm{Re}}(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}) \tag{6.3b}$$

$$\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}=-\frac{\partial p}{\partial z}+\frac{1}{\mathrm{Re}}(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}) \tag{6.3c}$$

$$\frac{\partial u}{\partial \mathrm{x}}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{6.3d}$$

where *u*, *v* and *w* are the velocity components along *x*-direction, *y*-direction and *z*-direction, respectively.

In order to solve the above Navier-Stokes equations, special splitting technique is required to deal with the difficulties arising from lack of an independent equation for the pressure, whose gradient contributes to the momentum equations. In this study, fractional step method proposed by Chorin (1968) is adopted to do the job.

In addition, the hybrid FD and local MQ-DQ method is used for spatial discretization.

6.3.3 Fractional step method

For convenient illustration of the fractional step method, governing equations (6.3a)-(6.3d) are written in vector form as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{\mathrm{Re}} \Delta \mathbf{u} \tag{6.4a}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{6.4b}$$

Solution of above N-S equations encounters difficulties like the lack of an independent equation for the pressure and non-existence of a dominant variable in the continuity equation. One way to circumvent these difficulties is to decouple the pressure computation from the momentum equations and then construct a pressure field so as to enforce the satisfaction of the continuity equation. In this work, a two step fractional step formulation is applied for 3D N-S equations with a collocated/ non-staggered arrangement. In this scheme, the solution is advanced from time level "*n*" to "*n+1*" through a predicting advection-diffusion step where pressure term is dropped from the momentum equations. In the advection-diffusion equations, convective and diffusive terms are discretized by using Crank-Nicolson scheme.

For a time increment $\Delta t = t^{n+1} - t^n$, the algorithm of fractional step consists of two steps: Firstly, an intermediate velocity $\mathbf{u}^*$ is predicted by the advection-diffusion equation, which drops the pressure term. That is, for each interior node in the domain, the intermediate velocity $\mathbf{u}^*$ can be calculated by

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \frac{1}{\mathrm{Re}}\frac{1}{2}(\Delta \mathbf{u}^* + \Delta \mathbf{u}^n) - \frac{1}{2}(\mathbf{u}^* \cdot \nabla \mathbf{u}^* + \mathbf{u}^n \cdot \nabla \mathbf{u}^n) \tag{6.5}$$

Secondly, the complete velocity $\mathbf{u}$ at t^{n+1} is corrected by including the pressure field, given by

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\nabla p^{n+1} \tag{6.6}$$

The final velocity field is subject to the continuity constraint given by

$$\nabla \cdot \mathrm{u}^{n+1} = 0 \tag{6.7}$$

Substituting equation (6.6) into equation (6.7) leads to the following Poisson equation for pressure

$$\Delta p^{n+1} = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t} \tag{6.8}$$

The velocity $\mathbf{u}^{n+1}$ is updated by the solution of pressure equation (6.8) and equation (6.6).

6.3.4 Implementation of Boundary conditions

The physical boundary conditions of the problem are specified as follows. Free stream boundary conditions (u = 1, v = w = 0) are applied at the inflow and transverse boundaries. Neumann boundary conditions are imposed at the outflow boundary. On the solid boundary, the no-slip boundary conditions ($u = v = w = 0$) are imposed.

Besides the above physical boundary conditions, some other boundary conditions are worthy of attention. One of them is the enforcement of continuity equation on the solid boundary. To achieve this, continuity equation should be accurately enforced on the solid boundary, i.e.

$$\frac{\partial(\mathbf{u}\cdot\mathbf{n})}{\partial n} = 0 \tag{6.9}$$

where n is the normal direction to the boundary surface.

For generality and consistency, the implementation of above Neumann boundary condition is carried out by using the so-called locally orthogonal “grid”. As shown in Shu et al. (2003), the reason for adopting this type of grid is for easy implementation. With the help of locally orthogonal “grid”, the Neumann boundary condition for the velocity components can be discretized by one-side finite difference scheme. Then, this discretized boundary condition is used to

update the corresponding velocity component at immediate interior point of the boundary node (not the boundary node itself), since the velocity on the boundary is already known.

The implementation of this boundary condition is shown below. For the convenience of illustration, the implementation procedure is shown in 2D space. The procedure in 3D space is similar except that one more vector should be considered. A 2D sketch is shown in Figure 6.4. For point 1, its original velocity vector is $\{u_1, v_1\}$ and its unit normal vector is $\{\frac{x}{r}, \frac{y}{r}\}$. Thus, $U_{n1} = u_1 \frac{x}{r} + v_1 \frac{y}{r}$. Then we can get the normal and tangent velocity vector to be $\{U_{n1} \frac{x}{r}, U_{n1} \frac{y}{r}\}$, $\{u_1 - U_{n1} \frac{x}{r}, v_1 - U_{n1} \frac{y}{r}\}$, respectively. Similarly, $U_{n2} = u_2 \frac{x}{r} + v_2 \frac{y}{r}$.

From the boundary condition $\frac{\partial(\mathbf{u} \cdot \mathbf{n})}{\partial n} = 0$, we get $\frac{\partial U_n}{\partial n} = 0$. Discretizing it by one-sided FD scheme, we have

$$\left.\frac{\partial U_n}{\partial n}\right|_0 = \frac{dr_2^2 \cdot U_{n1} - dr_1^2 \cdot U_{n2} - (dr_2^2 - dr_1^2) U_{n0}}{dr_2^2 \cdot dr_1 - dr_1^2 \cdot dr_2} = 0 \tag{6.10}$$

Rearranging the above equation, we can get

$$U_{n1} = \frac{(dr_2^2 - dr_1^2) U_{n0} + dr_1^2 \cdot U_{n2}}{dr_2^2} \tag{6.11}$$

Since $U_{n0} = 0$, U_{n1} can be simplified to be

$$U_{n1} = \frac{dr_1^2}{dr_2^2} U_{n2} \tag{6.12}$$

Thus, the normal velocity at point 1 is updated to be $U'_{n1} = \frac{dr_1^2}{dr_2^2} U_{n2}$ and its corresponding vector becomes $\{U'_{n1}\frac{x}{r}, U'_{n1}\frac{y}{r}\}$. Since the tangent velocity vector of point 1 remains unchanged, the new velocity vector is updated to be:

$$\{U'_{n1}\frac{x}{r}, U'_{n1}\frac{y}{r}\} + \{u_1 - U_{n1}\frac{x}{r}, v_1 - U_{n1}\frac{y}{r}\}$$

$$= \{u_1 + (U'_{n1} - U_{n1})\frac{x}{r}, v_1 + (U'_{n1} - U_{n1})\frac{y}{r}\} \tag{6.13}$$

From the above derivation, we can easily update the velocity at point 1 by

$$u'_1 = u_1 + (U'_{n1} - U_{n1})\frac{x}{r} \tag{6.14a}$$

$$v'_1 = v_1 + (U'_{n1} - U_{n1})\frac{y}{r} \tag{6.14b}$$

where $U_{n1} = u_1\frac{x}{r} + v_1\frac{y}{r}$, $U'_{n1} = \frac{dr_1^2}{dr_2^2} U_{n2}$ and $U_{n2} = u_2\frac{x}{r} + v_2\frac{y}{r}$.

The other concern is the boundary conditions for pressure equation. In general, there is no explicit boundary condition for pressure on the solid surface. However, it can be derived from the momentum equations at the boundary node. For the present case, the pressure gradient normal to the boundary at the boundary nodes can be expressed as

$$\frac{\partial p}{\partial n} = \nabla p \cdot \vec{n} = \begin{pmatrix} \frac{\partial p}{\partial x} & \frac{\partial p}{\partial y} & \frac{\partial p}{\partial z} \end{pmatrix} \cdot \begin{pmatrix} \frac{x}{r} \\ \frac{y}{r} \\ \frac{z}{r} \end{pmatrix} = \frac{\partial p}{\partial x}\frac{x}{r} + \frac{\partial p}{\partial y}\frac{y}{r} + \frac{\partial p}{\partial z}\frac{z}{r} \tag{6.15}$$

where $\frac{\partial p}{\partial x} = \frac{1}{\mathrm{Re}}\nabla^2 u$, $\frac{\partial p}{\partial y} = \frac{1}{\mathrm{Re}}\nabla^2 v$ and $\frac{\partial p}{\partial z} = \frac{1}{\mathrm{Re}}\nabla^2 w$

6.3.5 Solution Procedure

The solution procedure for applying the fractional step method to solve the 3D N-S equations is shown below:

(1) The advection-diffusion equation (6.5) is solved by SOR method to obtain the intermediate velocity $\mathbf{u}^*$. (Boundary conditions for the intermediate velocity are chosen to be the same as the physical boundary conditions).

(2) The pressure Poisson equation (6.8) is solved by SOR method to obtain the pressure for the next step.

(3) Update the pressure boundary conditions (6.15) on the solid boundary.

(4) Update the velocity in the domain for the next step through equation (6.6). (Velocity on the boundary is fixed)

(5) Enforce the continuity equation on the solid boundary to update the velocity near the boundary through equation (6.14a-6.14b).

(6) Go to step (1) until a solution for steady state is reached.

6.3.6 Calculation of drag coefficient C_D

For an incompressible viscous flow, we get the following constitutive relation

$$\tau = \mu(\nabla\vec{v} + \nabla\vec{v}^T) \tag{6.16}$$

where $\nabla\vec{v} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial w}{\partial z} \end{pmatrix}$ and $\nabla\vec{v}^T = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{pmatrix}$

Substituting matrices $\nabla\vec{v}$ and $\nabla\vec{v}^T$ into equation (6.16), we can get

$$\tau = \mu\begin{pmatrix} 2\frac{\partial u}{\partial x} & \frac{\partial v}{\partial x}+\frac{\partial u}{\partial y} & \frac{\partial w}{\partial x}+\frac{\partial u}{\partial z} \\ \frac{\partial u}{\partial y}+\frac{\partial v}{\partial x} & 2\frac{\partial v}{\partial y} & \frac{\partial w}{\partial y}+\frac{\partial v}{\partial z} \\ \frac{\partial u}{\partial z}+\frac{\partial w}{\partial x} & \frac{\partial v}{\partial z}+\frac{\partial w}{\partial y} & 2\frac{\partial w}{\partial z} \end{pmatrix} \tag{6.17}$$

With the above expression for the stress tensor, the traction vector can be derived as

$$\begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} = \tau \cdot \vec{n} = \mu \begin{pmatrix} 2\frac{\partial u}{\partial x} & \frac{\partial v}{\partial x}+\frac{\partial u}{\partial y} & \frac{\partial w}{\partial x}+\frac{\partial u}{\partial z} \\ \frac{\partial u}{\partial y}+\frac{\partial v}{\partial x} & 2\frac{\partial v}{\partial y} & \frac{\partial w}{\partial y}+\frac{\partial v}{\partial z} \\ \frac{\partial u}{\partial z}+\frac{\partial w}{\partial x} & \frac{\partial v}{\partial z}+\frac{\partial w}{\partial y} & 2\frac{\partial w}{\partial z} \end{pmatrix} \cdot \begin{pmatrix} \frac{x}{r} \\ \frac{y}{r} \\ \frac{z}{r} \end{pmatrix}$$
$$= \mu \begin{pmatrix} 2\frac{\partial u}{\partial x}\frac{x}{r}+(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y})\frac{y}{r}+(\frac{\partial w}{\partial x}+\frac{\partial u}{\partial z})\frac{z}{r} \\ (\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x})\frac{x}{r}+2\frac{\partial v}{\partial y}\frac{y}{r}+(\frac{\partial w}{\partial y}+\frac{\partial v}{\partial z})\frac{z}{r} \\ (\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x})\frac{x}{r}+(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y})\frac{y}{r}+2\frac{\partial w}{\partial z}\frac{z}{r} \end{pmatrix} \quad (6.18)$$

And the contribution due to the pressure can be written as

$$\begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix} = p\vec{n} = p\begin{pmatrix} \frac{x}{r} \\ \frac{y}{r} \\ \frac{z}{r} \end{pmatrix} = \begin{pmatrix} p\frac{x}{r} \\ p\frac{y}{r} \\ p\frac{z}{r} \end{pmatrix} \quad (6.19)$$

The drag force is the force in the *x*-direction, which can be written as

$$F_D = S(F_x - P_x) = S[\mu\{2\frac{\partial u}{\partial x}\frac{x}{r}+(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y})\frac{y}{r}+(\frac{\partial w}{\partial x}+\frac{\partial u}{\partial z})\frac{z}{r}\}-p\frac{x}{r}] \quad (6.20)$$

where F_D is the drag force, *S* represents the surface area of the sphere, F_x and P_x are the stress and pressure force in the *x*-direction, respectively.

With the above drag force expression, the drag force coefficient C_D is expressed as

$$C_D = \frac{F_D}{\frac{1}{2}\rho U_\infty^2 S'} = \frac{S[\mu\{2\frac{\partial u}{\partial x}\frac{x}{r} + (\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y})\frac{y}{r} + (\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z})\frac{z}{r}\} - p\frac{x}{r}]}{\frac{1}{2}\rho U_\infty^2 S'} \tag{6.21}$$

where $S' = \pi r^2$ represents the circle area.

6.3.7 Results and Discussion

The problem of flow past a sphere has been studied by Johnson and Patel (1999) in details. They found that when $20 \le \mathrm{Re} \le 210$, the flow is separated, steady, axisymmetric and topologically similar. And when $210 \le \mathrm{Re} \le 270$, although the flow still remains steady, it is non-axisymmetric. In this study, the flows with Reynolds numbers of 50, 100, 150 and 200 are studied as axisymmetric case and Reynolds number of 250 as a non-axisymmetric case.

As mentioned above, the hybrid FD and local MQ-DQ method is adopted to simulate this flow problem. In the local MQ-DQ method, 32 supporting points are employed for every reference node and the shape parameter is set to be 0.18, based on the observation in Section 6.2. Central difference scheme is applied on the uniform Cartesian mesh. The time step for temporal discretization is set to be

0.01. For the convergence criterion for steady flow, L_2 norm of velocity difference between the new and old time levels, i.e. $\left\| u^{n+1} - u^n \right\|$, is set to be less than 10^{-4}, which is considered to be small enough for a converged solution.

6.3.7.1 Steady axisymmetric flow

For flow in this regime, streamlines are shown in Figure 6.5, which displays the streamlines on the (x, y)-plane at $z = 0$ for Reynolds numbers of 50, 100, 150 and 200. In these and all the following figures, unless otherwise noted, the flow direction is from left to right. From Figure 6.5, it can be seen that the flow separates from the surface of the sphere at a separation angle and rejoins at a point on the axis of the flow to form a closed separation bubble and a toroidal vortex. It can also be seen from Figure 6.5 that the flow structure for all the four Reynolds numbers remains topologically the same with changes only in the separation angle, the vortex's position and the separation bubble length. To illustrate the accuracy of the present results, the separation bubble length L_{sep} together with the drag coefficient C_d obtained by the hybrid method is compared with those by Johnson and Patel (1999). Table 6.3 shows good agreement between the present results and the aforementioned results.

Pressure contours for Reynolds numbers of 50, 100, 150 and 200 are shown in Figure 6.6. Contours are drawn for every 0.05 increment with dashed lines used

for negative values. By comparing the pressure contours, we can see that, within the wake, in the vicinity of the toroidal vortex shown in Figure 6.5, there is no pressure minimum in the symmetry plane until a Reynolds number of 200. As shown in Figure 6.6(d), the closed circles in the wake indicate a pressure minimum located very near the center of the toroidal vortex.

6.3.7.2 Steady non-axisymmetric flow

As has been pointed out by Johnson and Patel (1999), at a Reynolds number of 211, the calculated flow solution no longer exhibits axial symmetry. The flow does, however, remain steady. Although non-axisymmetric, the flow does contain a plane of symmetry. The extent of this steady non-axisymmetric range has been documented as approximately $210 < \mathrm{Re} < 270$. The results within this regime were found to be essentially self-similar, or topologically identical, thus only the solution at $\mathrm{Re} = 250$ is considered in the paper of Johnson and Patel (1999) since the departure from symmetry is quite pronounced. In this study, we also only consider the case with *Re* = 250 for the convenience of comparison.

Figures 6.7(a) and 6.7(b) show streamlines in the (x, z)- and (x, y)- planes, respectively. It is clear from Figure 6.7(a) that the flow field is symmetric about the (x, z)-plane, which divides the figure across the center. Comparatively, from Figure 6.7(b), it is apparent that the toroidal vortex has tilted. It is also clear from

the difference between the top and bottom of the vortex ring, that its size is not constant in the azimuthal direction. Additionally, the toroid is clearly no longer a closed separation bubble. The upper spiral is actually fed by fluid originating from upstream while the lower spiral releases fluid into the wake after sending it up and around the upper spiral.

Pressure contours for the (x, z)- and (x, y)-planes are shown in Figure 6.8. Like Figure 6.6, the contours are drawn in levels of 0.05. The pressure field in the (x, z)-plane is completely symmetric and closely resembles the contours in Figure 6.6(d) for a Reynolds number of 200, although the pressure in the core of the vortex is clearly lower owing to the higher centrifugal acceleration of the vortex. Pressure contours in the (x ,y)-plane are not symmetric. The pressure minimum in the region of the lower focus of Figure 6.7(b) is lower than that in the region of the upper focus. This azimuthal pressure gradient propagates through the core of the toroidal vortex inducing flow along the vortex axis. This breakdown in axial symmetry, which begins at $\mathrm{Re} \approx 211$, corresponds closely to the occurrence of a global pressure minimum in the center of the vortex. It appears, therefore, that the instability of the axisymmetric flow is connected to the generation, by radial acceleration around the vortex center, of a ring of low pressure in the wake.

6.4 Lid-driven flow in a cubic cavity with a stationary, rigid sphere at its centre

In Section 6.3, the capability and flexibility of the 3D local MQ-DQ method to solve fluid flow problems with curved boundary is proved. In this section, we will use the proven 3D local MQ-DQ method to study another 3D problem: lid-driven flow in a cubic cavity with a stationary, rigid sphere at its centre. This case has been simulated by Gilmanov et al (2003) to test the spatial accuracy of their new algorithm. However, details of the fluid flow phenomenon are not discussed. In this section, this problem will be studied in detail.

We will use the same geometry as the one used by Gilmanov et al (2003) for the convenience of comparison. A sphere is located at the centre of a cubic cavity. The diameter of the sphere is half of the cavity height. Flow inside the cavity is driven by the moving lid. In order to avoid corner singularities, the lid velocity is prescribed as follows:

$$U_{Lid}(x) = \begin{cases} 0.5\left[1 + \sin\left(4\pi x - \frac{\pi}{2}\right)\right] & x \leq 0.25 \\ 1 & 0.25 < x < 0.75 \\ 0.5\left[1 + \sin\left(4\pi x - \frac{\pi}{2}\right)\right] & x \geq 0.75 \end{cases} \quad (6.22)$$

The physical boundary conditions of the problem are specified as:

u=0, v=0, w=0 on the (x-y) plane with z=0 or z=1

u=0, v=0, w=0 on the (y-z) plane with x=0 or x=1

u=0, v=0, w=0 on the (z-x) plane with y=0

$u = U_{Lid}(x)$, v=0, w=0 on the (z-x) plane with y=1

u=0, v=0, w=0 on the surface of the sphere

Numerical simulation is firstly conducted with Reynolds number of 20 (based on the sphere diameter and the lid maximum velocity). Obtained results are validated by the results of Gilmanov et al (2003). Then flows with Reynolds numbers of 50, 100 and 200 are simulated to study the flow patterns at different Reynolds numbers. To conduct the simulation, we use the hybrid numerical method mentioned in Section 6.3. Grid points generated are shown in Figure 6.9. Only the points on the (x-y) plane with z=0.5 are shown in the figure for the convenience of illustration. From the figure, it can be seen that the mesh size of the Cartesian mesh is 0.02. In the local MQ-DQ method, 32 supporting points are employed for each reference node and the shape parameter is set to be 0.18. The time step for temporal discretization is set to be 0.005. For the convergence criterion of steady flow, L_2 norm of velocity difference between the new and old time levels, i.e. $\left\| u^{n+1} - u^n \right\|$, is set to be less than 5×10^{-6}, which is considered to be small enough for a converged solution.

To validate our results, we take the results by Gilmanov et al (2003) as reference data. Since this flow problem was only taken to test the accuracy of the algorithm by Gilmanov et al (2003), its flow pattern is not illustrated in details. Only the streamlines on the (x-y) plane at z=0.5 are shown. Thus the streamlines on the (x-

y) plane at z=0.5 from our results will be compared with those of Gilmanov et al (2003). The streamlines are shown in Figure 6.10. From Figure 6.10, it can be seen that there is a vortex between the moving lid and the sphere. The position of the vortex center is compared between our result and the result by Gilmanov et al (2003) in Table 6.4. It can be seen from Table 6.4 that our result agrees well with that by Gilmanov et al (2003).

After the validation of the results by the local MQ-DQ method, the details of this flow problem with Reynolds numbers of 20, 50, 100 and 200 are studied. To show the patterns of the flow field, three centroidal planes of the cube, which are located at x=0.5, y=0.5 and z=0.5, respectively, are chosen. Two dimensional planar projections of the velocity vector field at Re=20, 50, 100 and 200 on the (x-y) plane are shown in Figure 6.11. Pressure contours of these Reynolds numbers on the (x-y) plane are shown in Figure 6.12. These figures illustrate the change of flow patterns and pressure contours as Reynolds number varies. It can be observed from the flow pattern in the plane of z=0.5 (Figure 6.11) that the vortex area on top of the sphere changes from relatively flat to circular shape and the axis of the vortex starts in the upper right side to the sphere, then gradually moves towards the upper right corner of the box as the Reynolds number increases. It can also be observed from Figure 6.11 that the streamline pattern changes significantly with the increase of the Reynolds number, especially the streamlines above the spherical surface. The change of flow patterns and pressure contours in the planes

of x=0.5 and y=0.5 are shown in Figures 6.13 & 6.14 and Figures 6.15 & 6.16, respectively.

6.5 Conclusions

In this Chapter, the local MQ-DQ method was extended to solve fluid flow problems with curved boundary in three-dimensional space. An error estimate was provided for the 3D local MQ-DQ method to study the influence of the number of supporting points and shape parameter on its numerical accuracy. It was observed that the accuracy of numerical solutions can be improved by increasing the value of shape parameter and the convergence rate can be improved by increasing the number of supporting points. Based on these findings, the local MQ-DQ method is then applied to solve three-dimensional, time-dependent, incompressible Navier-Stokes equations in the primitive variable form. The problem of flow past a sphere was simulated to demonstrate its capability and flexibility in solving 3D fluid flow problems with curved boundary. A hybrid FD and local MQ-DQ method was adopted to improve the efficiency of numerical computation. The obtained numerical results were compared well with data in the literature. Moreover, a lid-driven flow in a square cavity with a stationary, rigid sphere at its center was studied by the hybrid method. The numerical experiments show that the local MQ-DQ method is a promising scheme for solving 3D fluid flow problems with curved boundary.

Table 6.1 Mean value of convergence rate with

shape parameter $c = 0.2$

Number of supporting points	$T = \sin(\pi x)\sin(\pi y)\sin(\pi z)$	$T = x^4 + y^4 + z^4$
$n_s = 6$	2.06	1.97
$n_s = 18$	2.02	1.91
$n_s = 26$	2.00	1.89
$n_s = 30$	1.97	2.32
$n_s = 31$	1.94	2.20
$n_s = 32$	3.95	3.87
$n_s = 36$	3.92	3.95

Table 6.2 Mean value of convergence rate with number of supporting points $n_s = 32$

Shape parameter	$T = \sin(\pi x)\sin(\pi y)\sin(\pi z)$	$T = x^4 + y^4 + z^4$
$c = 0.1$	3.61	3.53
$c = 0.15$	3.86	3.77
$c = 0.2$	3.95	3.87
$c = 0.25$	3.99	3.91
$c = 0.3$	3.98	3.88

Table 6.3 Comparison of recirculating length and drag coefficient

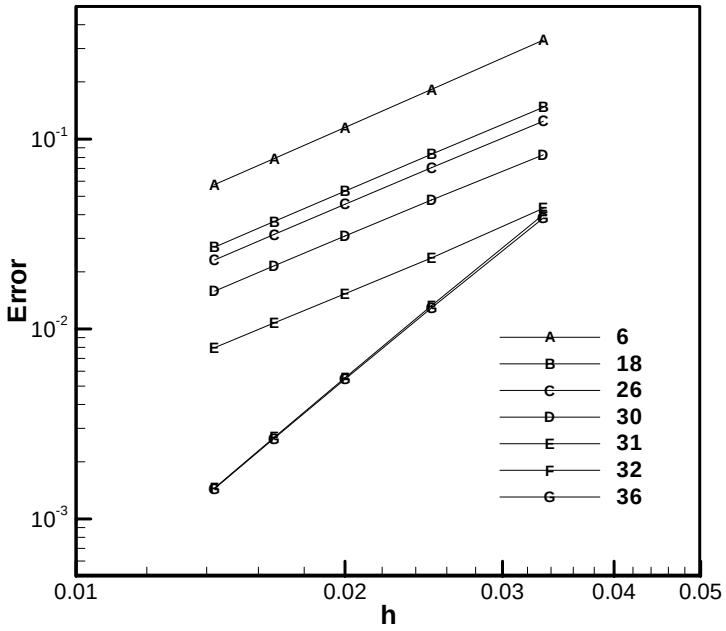
for Re = 50, 100, 150 and 200

Re	Source	L_{sep}	C_d
50	Present	0.413	1.564
	Johnson et al. (1999)	0.418	1.583
100	Present	0.896	1.103
	Johnson et al. (1999)	0.884	1.112
150	Present	1.292	0.901
	Johnson et al. (1999)	1.214	0.908
200	Present	1.552	0.786
	Johnson et al. (1999)	1.463	0.798

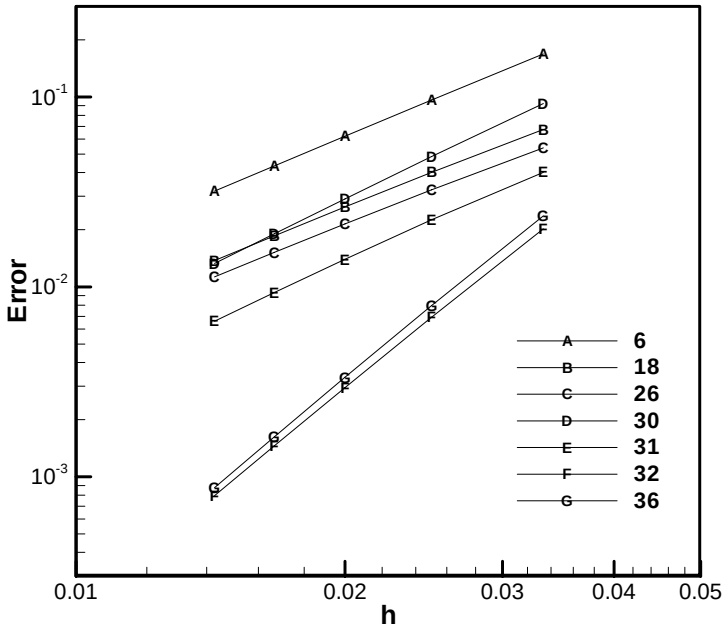
Table 6.4 Comparison of the position of vortex centre of

lid-driven cavity flow with Re=20

	x	**Y**
Gilmanov et al (2003)	0.661	0.822
Present	0.632	0.815



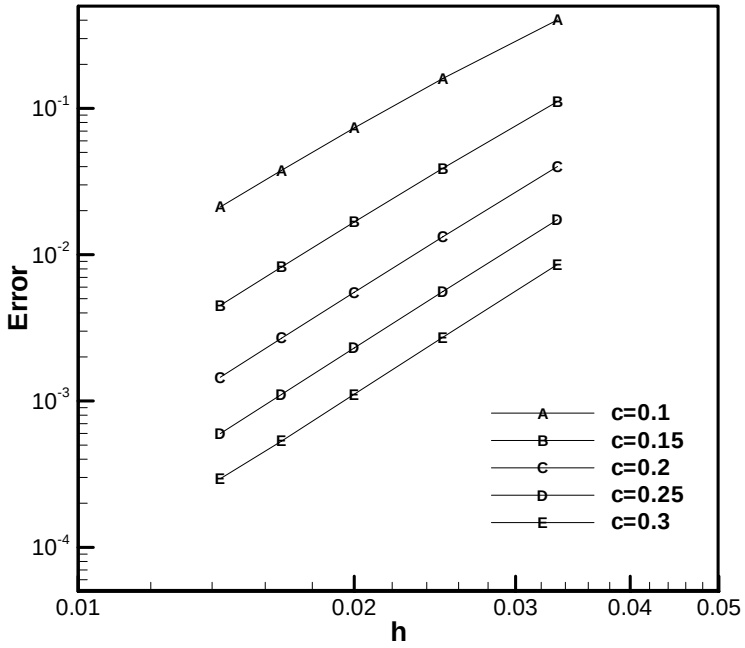
(a) $T = \sin(\pi x)\sin(\pi y)\sin(\pi z)$



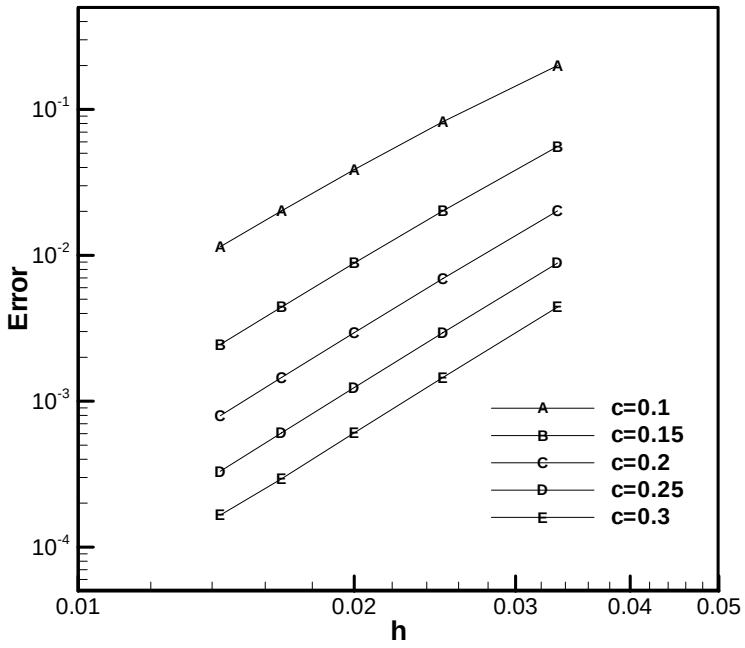
(b) $T = x^4 + y^4 + z^4$

Figure 6.1 Numerical errors versus mesh size

for various number of supporting points



(a) $T = \sin(\pi x)\sin(\pi y)\sin(\pi z)$



(b) $T = x^4 + y^4 + z^4$

Figure 6.2 Numerical errors versus mesh size for various shape parameter *c*

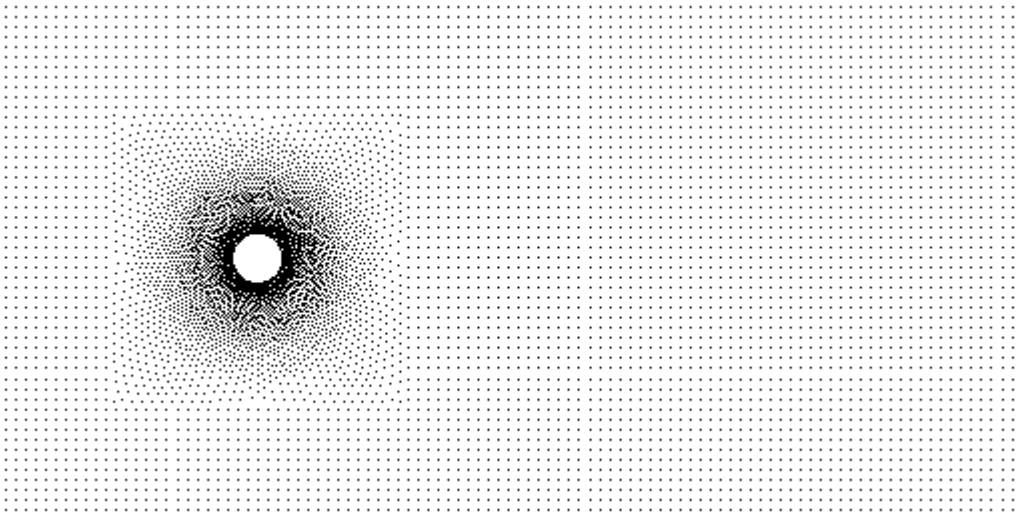


Figure 6.3 Grid point distribution on the *x*-*y* plane at $z = 0$

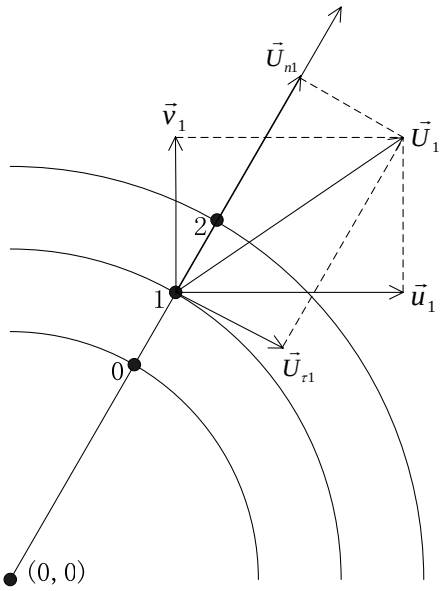
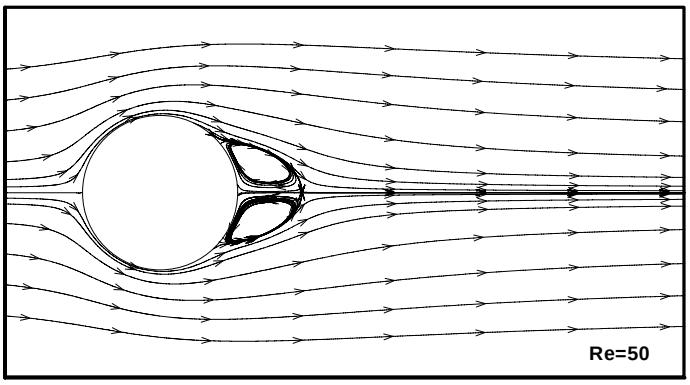
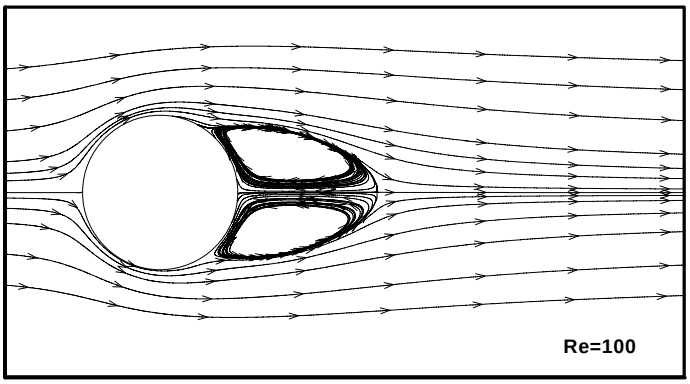


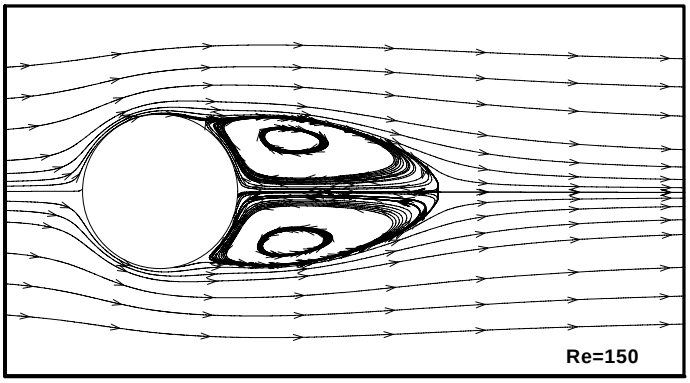
Figure 6.4 Sketch of the enforcement of continuity equation on the solid boundary



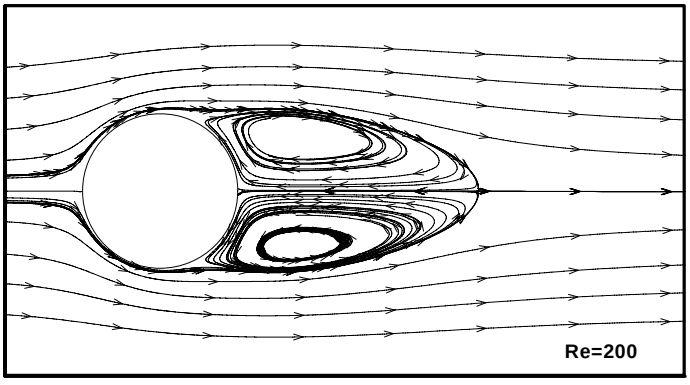
(a)



(b)

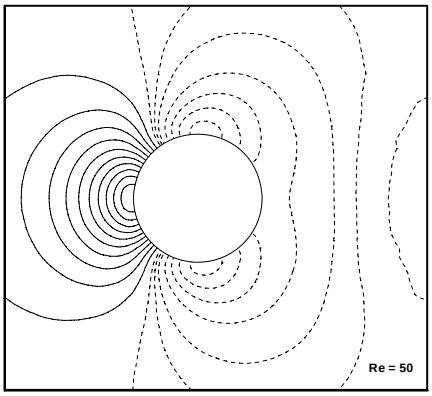


(c)

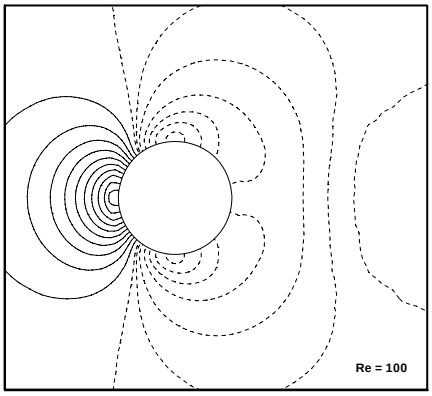


(d)

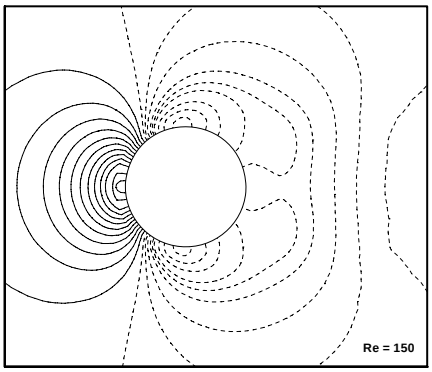
Figure 6.5 Calculated axisymmetric streamlines past the sphere



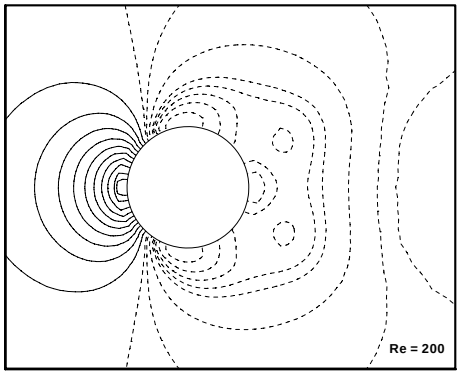
(a)



(b)

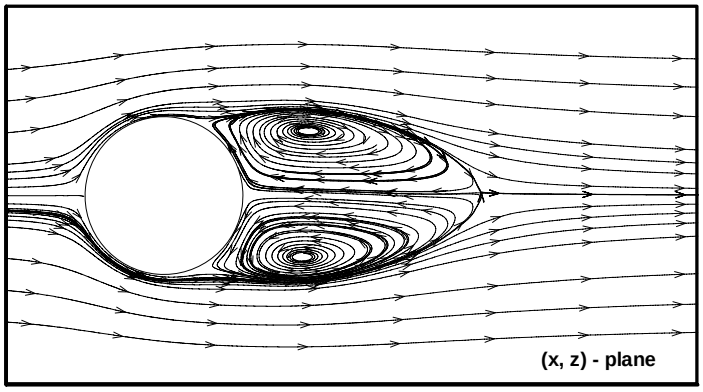


(c)

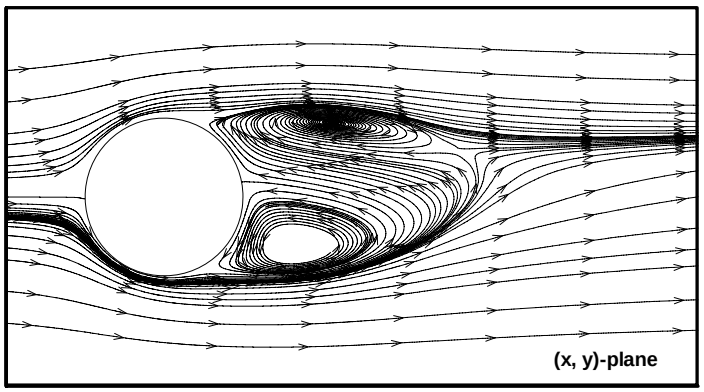


(d)

Figure 6.6 Pressure contours for axi-symmetric flow

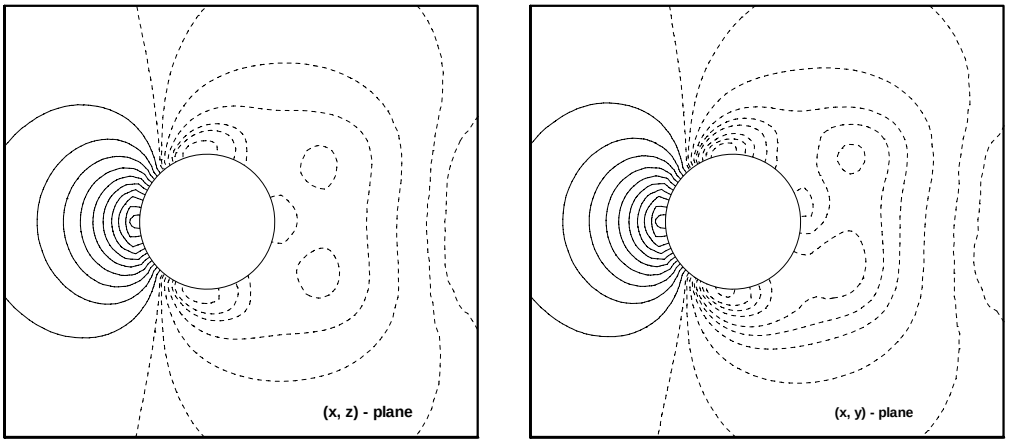


(a)



(b)

Figure 6.7 Streamlines of projected velocity vectors at Re = 250



(a) (b)

Figure 6.8 Pressure contours Re = 250

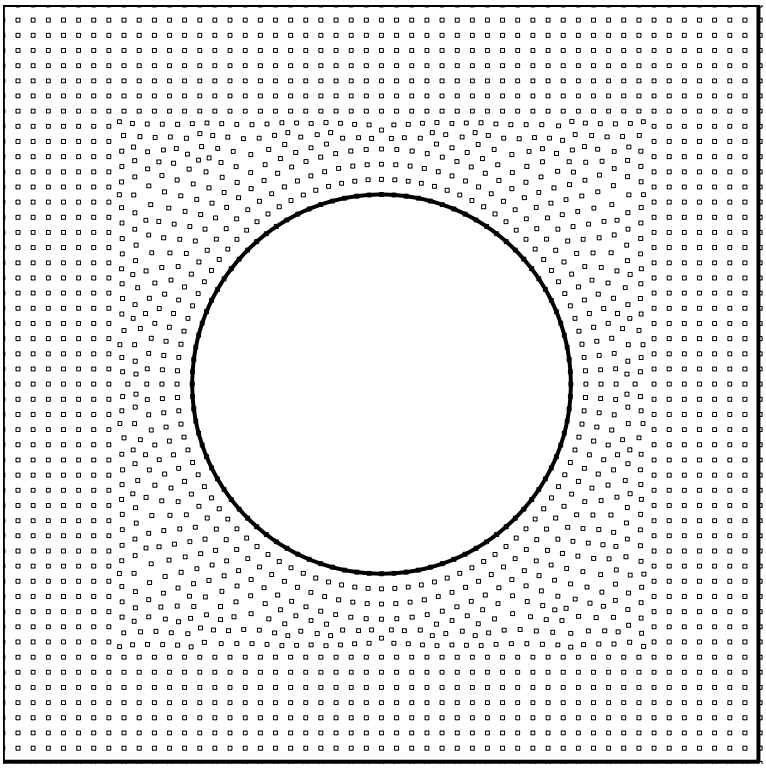
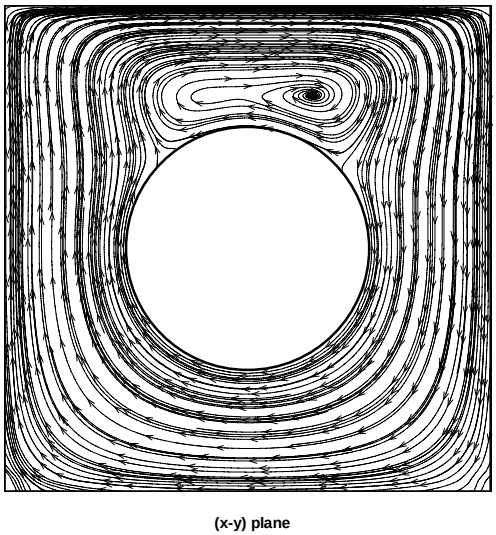
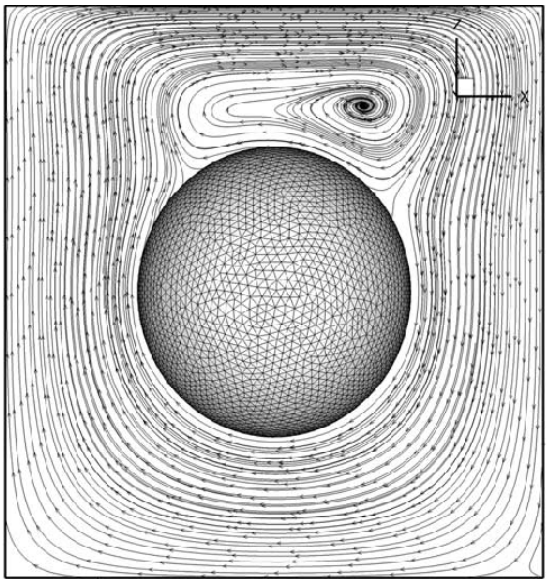


Figure 6.9 Grid points on (x-y) plane with z=0



(a) Results by the 3D local MQ-DQ method



(b) Results by Gilmanov et al (2003)

Figure 6.10 Streamlines on (x-y) plane at z=0

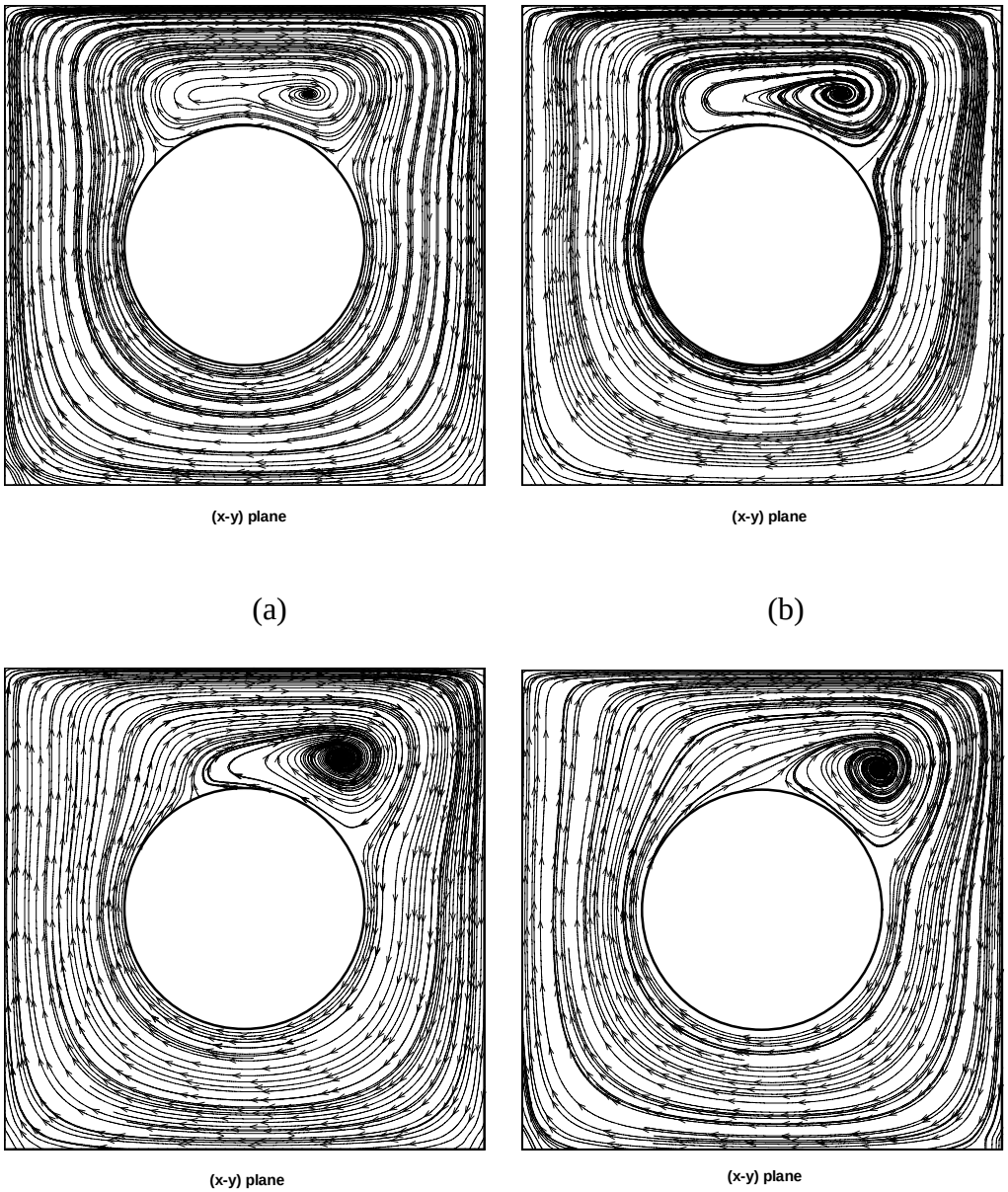
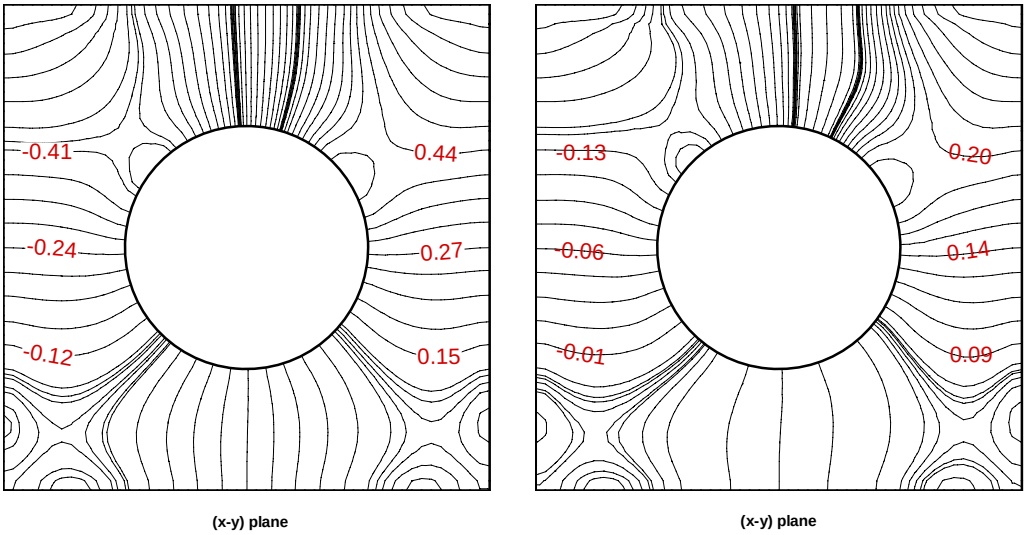
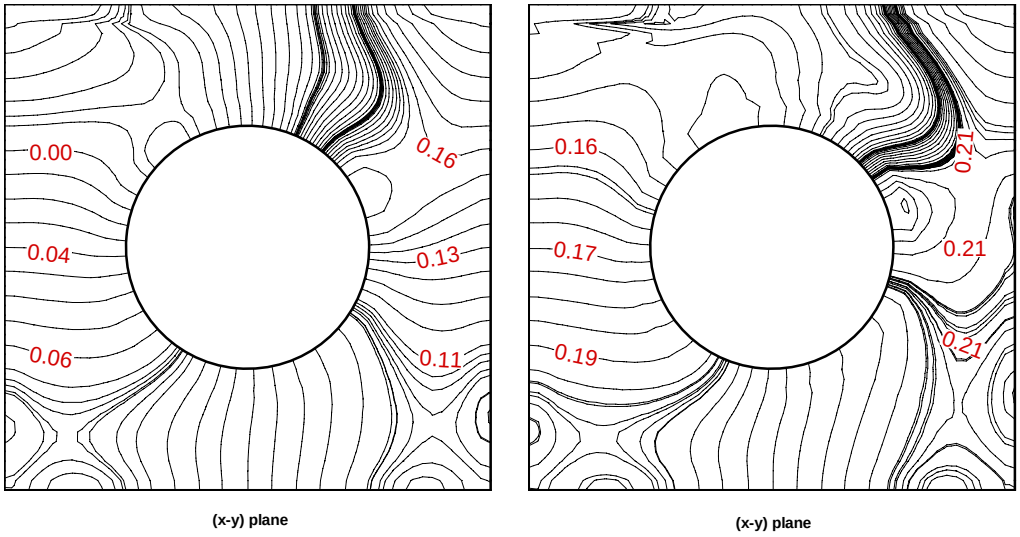


Figure 6.11 Streamlines of projected velocity vectors on (x-y) plane

(a) Re=20 (b) Re=50 (c) Re=100 (d) Re=200



(a) (b)



(c) (d)

Figure 6.12 Pressure contours on (x-y) plane

(a) Re=20 (b) Re=50 (c) Re=100 (d) Re=200

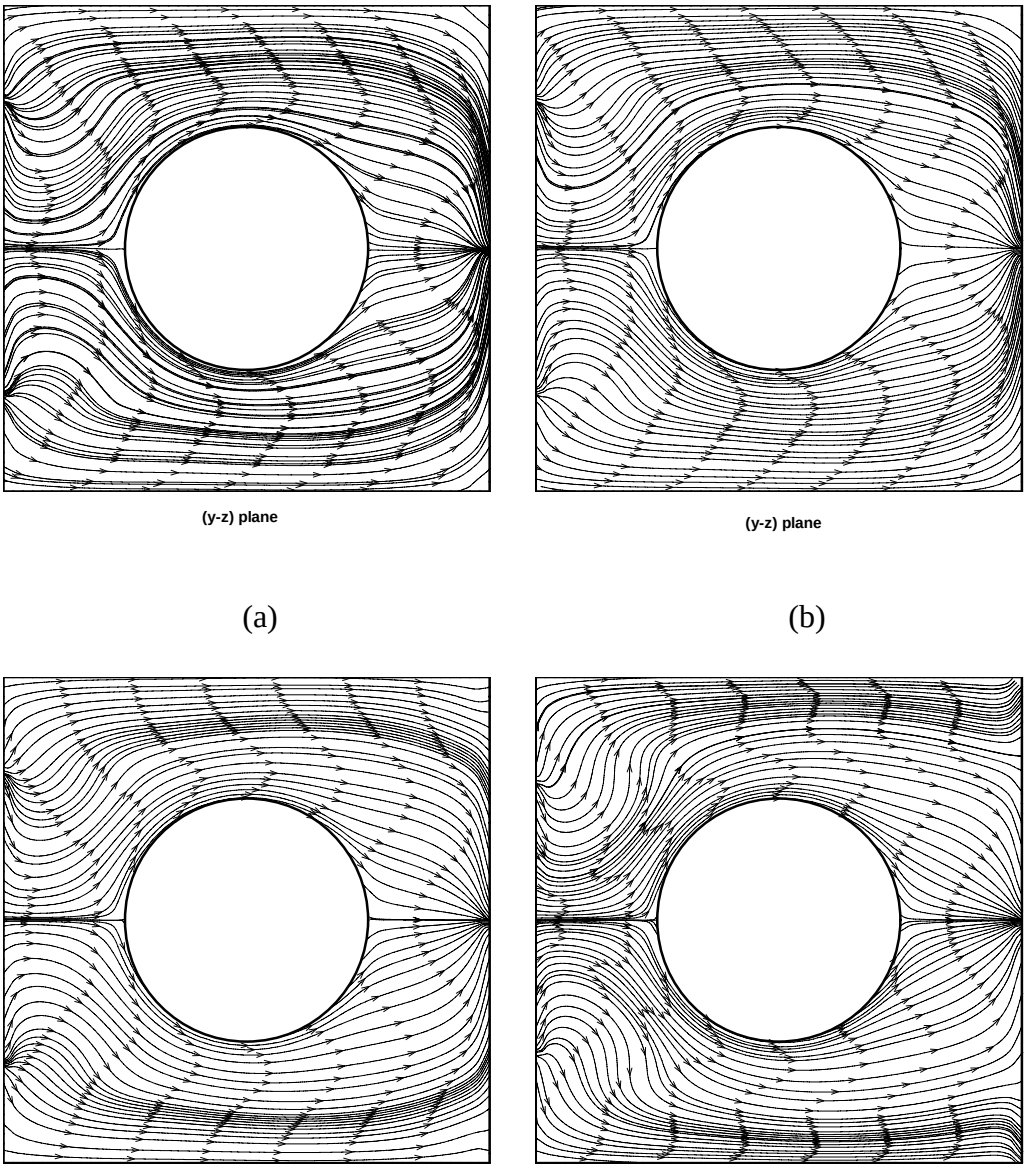
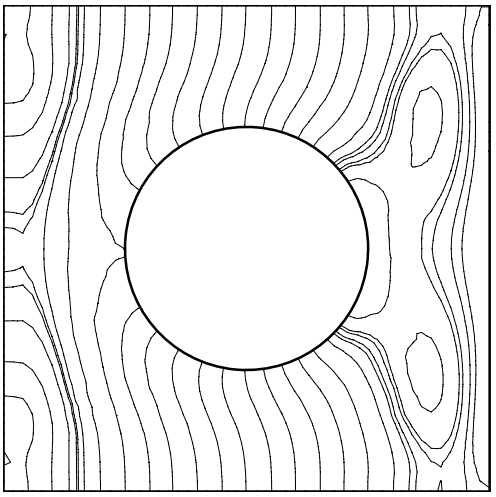
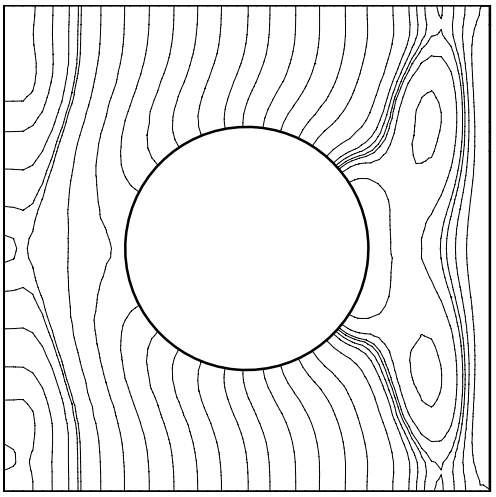


Figure 6.13 Streamlines of projected velocity vectors on (y-z) plane



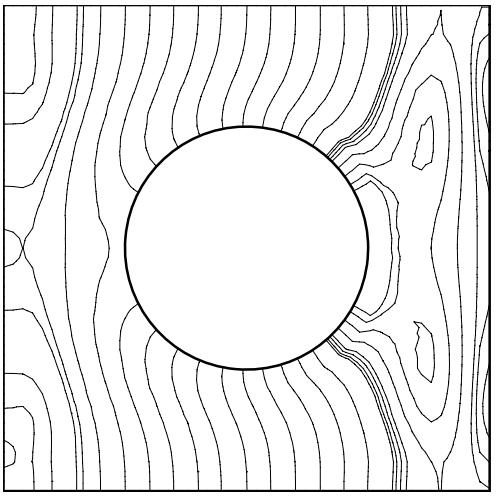
(y-z) plane

(a)



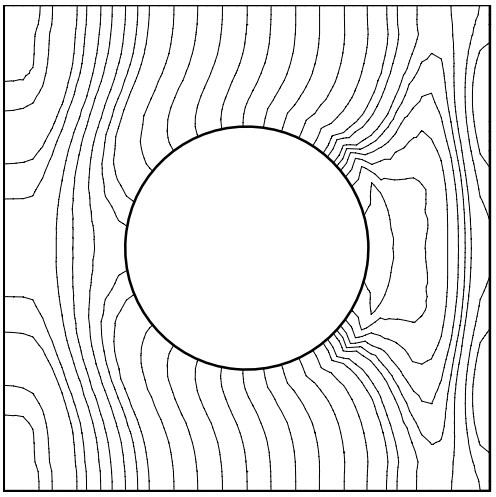
(y-z) plane

(b)



(y-z) plane

(c)



(y-z) plane

(d)

Figure 6.14 Pressure contours on (y-z) plane

(a) Re=20 (b) Re=50 (c) Re=100 (d) Re=200

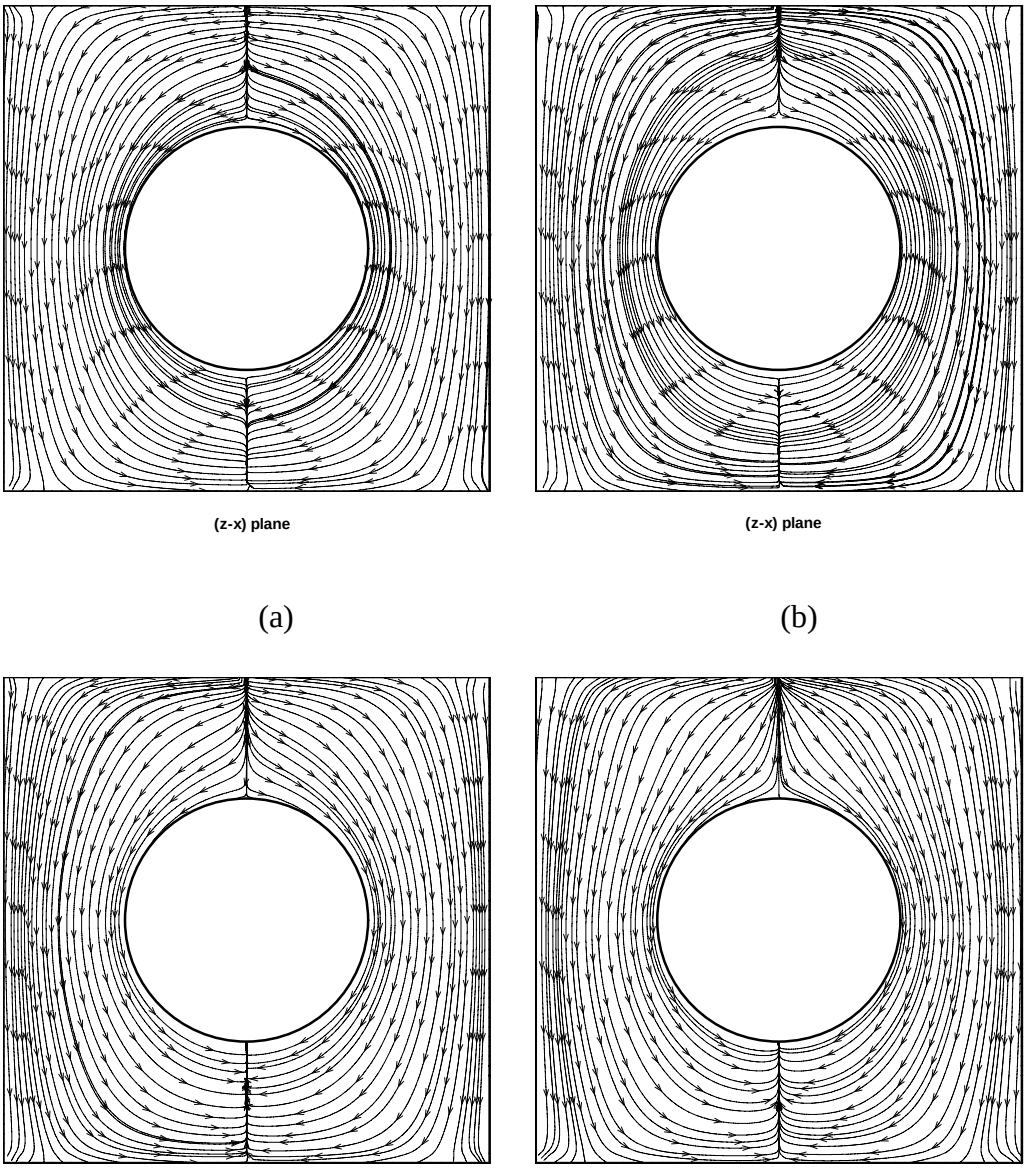
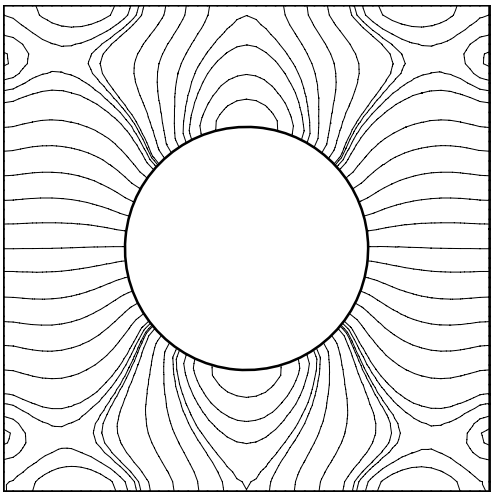


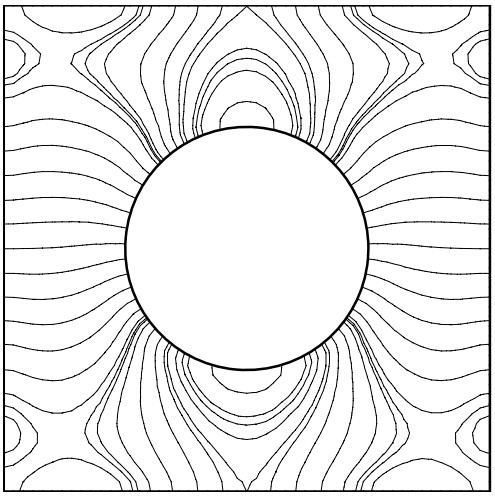
Figure 6.15 Streamlines of projected velocity vectors on (z-x) plane

(a) Re=20 (b) Re=50 (c) Re=100 (d) Re=200



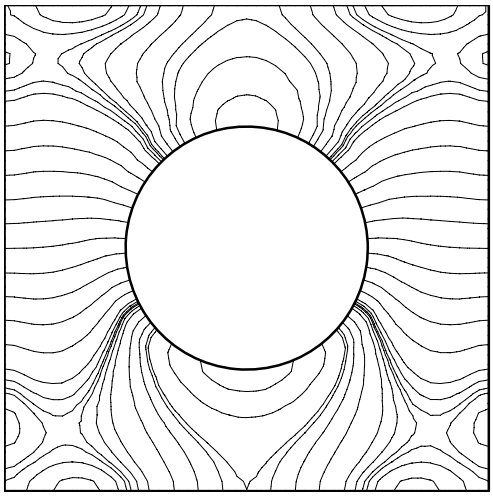
(z-x) plane

(a)



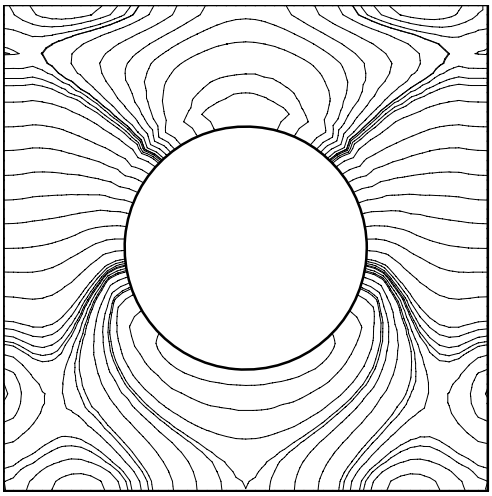
(z-x) plane

(b)



(z-x) plane

(c)



(z-x) plane

(d)

Figure 6.16 Pressure contours on (z-x) plane

(a) Re=20 (b) Re=50 (c) Re=100 (d) Re=200

Chapter 7 Conclusions and Recommendations

7.1 Conclusions

This thesis mainly focused on the study of the local MQ-DQ method. Some important issues of this method have been studied in detail. Firstly, we derived the formulas of the local MQ-DQ method with the stencils of central difference scheme and referred them as MQ-FD method. The effect of the shape parameter c in MQ on the formulas of the MQ-FD method was analyzed. The resultant formulas were compared with those of the central difference scheme. It was found that when c goes to infinity, the MQ-FD formulas of derivative approximation are the same as those given by the central difference scheme. This observation created a relationship between the MQ-FD method and the conventional FD scheme. The performance of the MQ-FD method for derivative approximation and solution of partial differential equations was systematically studied. It was found that if the shape parameter c is properly chosen, the MQ-FD method may solve periodic boundary value problems more accurately than the central difference scheme does. For general boundary value problems, however, the accuracy by the MQ-FD method may not be as accurate as that by the central FD scheme. When the value of c is not very small, the accuracy by these two methods is very close.

Secondly, an efficient local stencil adaptive algorithm was proposed for the application of the local MQ-DQ method to simulate two-dimensional fluid flow

problems with curved boundaries. In this algorithm, a body-fitted grid is initially generated as the background mesh. During the simulation, this algorithm is able to automatically adjust the scale of the local stencils based on the gradient of the solution. Thus, fine stencils are generated in regions where solutions are varying rapidly for better accuracy and resolution and coarse stencils are generated for regions where solutions are less active for better efficiency. The local MQ-DQ method was used to discretize the governing equations because of its mesh-free property. This adaptive local MQ-DQ method was validated by two numerical experiments with curved boundaries. Numerical results showed that this method can effectively solve problems with curved boundaries. Furthermore, it can solve problems as accurately as the local MQ-DQ method does on a regular grid, but with less grid points and running time. As a result, this local MQ-DQ based stencil adaptive method offers a promising approach to solve engineering problems with curved boundaries.

Thirdly, due to the low efficiency of the local MQ-DQ method, a hybrid technique was proposed to improve the efficiency of the numerical scheme while maintaining its flexibility for dealing with curved boundaries. This hybrid technique combined the local MQ-DQ method with the conventional FD scheme. The whole computational domain was divided into two parts: one small part near curved boundaries filled with randomly distributed grid points and the remaining larger part filled with Cartesian mesh points. The local MQ-DQ method was

applied on the randomly distributed grid points, taking its mesh-free property to deal with curved boundaries. And the conventional FD scheme was applied on the Cartesian mesh points, taking its high efficiency property to reduce computational costs. There was no information exchange requirement between the two parts of the whole domain. Thus, the application of the hybrid technique is quite simple and straightforward. This hybrid technique was adopted to simulate fluid flow problems both in 2-D and 3-D space. It was validated by simulating flow past a circular cylinder at both steady and unsteady states in 2D space, followed by the simulation of flow past a sphere in 3D space. The obtained results were in line with other computational and experimental data available in the literature. In addition, the results showed that the hybrid technique greatly improved the computational efficiency as compared with the fully local MQ-DQ method.

Finally, we studied the 3-D local MQ-DQ method and explored its ability to simulate fluid flow problems with curved boundaries in 3-D space. An error estimate was provided for the 3-D local MQ-DQ method to study the influence of the number of supporting points and shape parameter on its numerical accuracy. It was found that the accuracy of numerical solutions can be improved by increasing the value of shape parameter and the convergence rate can be improved by increasing the number of supporting points. An empirical relationship was obtained between the convergence rate and the number of supporting points. More specifically, it can be written as

$$\|\varepsilon\| \sim O(h^n) \quad \text{and} \quad n \approx \begin{cases} 2.0 & \textit{for } 6 \le \mathbf{n}_s \le 31 \\ 3.9 & \textit{for } 32 \le \mathbf{n}_s \le 36 \end{cases}$$

Based on these findings, the 3-D local MQ-DQ method was applied to simulate the flow past a sphere to demonstrate its capability and flexibility in solving 3-D fluid flow problems with curved boundaries. Some special techniques were proposed to deal with the boundary conditions on the curved boundaries. The obtained numerical results compared well with data in the literature. The numerical experiments showed that the local MQ-DQ method is a promising scheme for solving 3-D fluid flow problems with curved boundaries. Since up to now very little work has been done on the research of 3-D RBF-related methods, this work can be regarded as a pioneer effort. It may provide some useful guidance for further study of 3-D RBF-related methods.

In general, this study revealed some important and basic properties of the mesh-free local MQ-DQ method, such as its relationship with conventional FD scheme, the influence of the shape parameter *c* on the accuracy and convergence rate of numerical solutions and the determination of the shape parameter *c*. These findings seem to be able to serve as a useful guideline for further study of RBF-related mesh-free methods. Furthermore, our study made the local MQ-DQ method more practical and flexible for engineering applications. It was proven that both the local stencil adaptive technique and the hybrid technique can significantly improve the efficiency of this mesh-free method. It was also proven that the local MQ-DQ method is able to solve 3-D fluid flow problems with

complex geometries. Its flexibility for dealing with complex geometries and capability for 3-D flow problems together with the improvement of efficiency make this mesh-free method more attractive for solving complicated industrial problems.

7.2 Recommendations on future work

Despite the improvement of the local MQ-DQ method, there are still a lot of problems which need further study. First, the choice of the free shape parameter c is still an open problem for all the researchers in this field. Inappropriate choice of shape parameter c may cause stability problems or result in inaccurate solutions. Uncertainty on choosing shape parameter c makes the application of the local MQ-DQ method more difficult, especially for beginners. A better way to determine the shape parameter is very important for better development of this method. Second, the order of accuracy for this method has not been theoretically studied in this thesis. More work should be done to further study the order of accuracy for this method so that it can provide a basic idea for the users to choose their desirable order of accuracy. Moreover, the local MQ-DQ method may become unstable when the distribution of grid points is very uneven. Thus, future work should focus on how to solve this problem. There are two possible ideas which may tackle this problem. The first one is to implement the so-called least squares technique on the local MQ-DQ method. The least squares technique may make the local MQ-DQ method stable even with singularly distributed grid points.

The other one is to find a good way to generate well distributed grid points, thus making the method stable.

Bibliography

Anderson R. W., Elliott N. S. and Pember R. B., An arbitrary Lagrangian–Eulerian method with adaptive mesh refinement for the solution of the Euler equations, Journal of Computational Physics, 199 (2004) 598-617.

Basebi T. and Thomas R.M., A study of moving mesh methods applied to a thin flame propagating in a detonator delay element, Computers & Mathematics with Applications, 45 (2003) 131-163.

Beckett G., Mackenzie J.A., Ramage A. and Sloan D. M., Computational Solution of Two-Dimensional Unsteady PDEs Using Moving Mesh Methods, Journal of Computational Physics, 182 (2002) 478-495.

Behr M., Hastreiter D., Mittal S., Tezduyar T.E., Incompressible flow past a circular cylinder: dependence of the computed flow field on the location of the lateral boundaries, Comput. Methods Appl. Mech. Engng. 123 (1995) 309-316.

Bellman R., Casti J., Differential quadrature and long term integration, J. Math. Anal. Appl. 34 (1971) 235-238.

Bellman R., Kashef B.G., Casti J., Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations, J. Comput. Phys. 10 (1972) 40-52.

Berger M.J. and Leveque R.J., Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems, SIAM J. Numer. Anal. 35 (1998)

2298-2316.

Berger M.J. and Oliger J., Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys. 53 (1984) 484-512.

Braza M., Chassaing P., Minh H.H., Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, J. Fluid Mech. 165 (1986) 79.

Budd C.J., Carretero-Gonzalez R. and Russell R .D., Precise computations of chemotactic collapse using moving mesh methods, J. Comput. Phys., 202 (2005) 463-487.

Carlson R.E., Foley T.A., The parameter R^2 in multiquadric interpolation, Computer Math. Applic. 21 (1991) 29-42.

Chen W., New RBF collocation schemes and kernel RBFs with applications. Lecture Notes in Computational Science and Engineering, 26 (2002) 75-86.

Cheng AH-D, Golberg M.A., Kansa E.J., Zammito G., Exponential convergence and H-c multiquadric collocation method for partial differential equations. Numerical Methods for Partial Differential Equations, 19 (2003) 571–594.

Chorin, A.J., Numerical solution of the Navier-Stokes equations, Math. Comput., 22 (1968) 745-762.

Dennis S.C.R., Chang G.Z., Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100, J. Fluid Mech. 42 (1970) 471-489.

Ding H., Shu C., A stencil adaptive algorithm for finite difference solution of incompressible viscous flows, J. Comput. Phys. 214 (2006), 397-420.

Ding H., Shu C., Yeo K.S., Lu Z.L., Simulation of Natural Convection in Eccentric Annuli Between A Square Outer Cylinder and a Circular Inner Cylinder Using Local MQ-DQ Method, Numerical Heat Transfer, 47 (2005) 291-313.

Ding H., Shu C., Yeo K.S., Xu D., Simulation of incompressible viscous flows past a circular cylinder by hybrid FD scheme and meshless least square-based finite difference method, Comput. Methods Appl. Mech. Engrg. 193 (2004) 727-744.

Ding H., Shu C., Tang D.B., Error Estimates of Local Multiquadric-based Differential Quadrature (LMQDQ) Method through Numerical Experiments, Int. J. Numer. Meth. Engng, 63 (2005) 1513-1529.

Ding H., Shu C., Yeo K.S., Xu D., Numerical computation of three-dimensional incompressible viscous flows in the primitive variable form, Comput. Methods Appl. Mech. Engrg. 195 (2006) 516-533.

Driscoll T. A., Fornberg B., Interpolation in the limit of increasingly flat radial basis functions, comput. Math. Appl. 43 (2002) 413-422.

Dubal M.R., Olivera S.R. and Matzner R.A., Approaches to Numerical Relativity, Cambridge University Press, Cambridge, UK, 1993.

Duchon, J., Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces, RAIRO Analyse Numerique, 10 (1976) 5-12.

Eiseman Peter R., Adaptive grid generation, Computer methods in applied mechanics and engineering, 64 (1987) 321-376.

Fasshauer G.E., Solving partial differential equations by collocation with radial basis functions, in: A.L. Mehaute, C. Rabut, L.L. Schumaker (Eds.), Surface Fitting and Multiresolution Methods (1997), 131-138.

Fornberg B., Wright G., Stable computation of multiquadric interpolants for all values of the shape parameter, Computers & Mathematics with Applications, 48 (2004) 853-867.

Franke C., Schaback R., Solving partial differential equations by collocation using radial basis functions, Appl. Math. Comput. 93 (1998) 73-82.

Franke R., Scattered data interpolation: test of some methods, Math. Comput. 38 (1982) 181-200.

Ghia U., Ghia K.N. and Shin C.T., High-Re solutions for incompressible flow using the Navier-Stokes equations and a multi-grid method, Journal of Computational Physics, 48 (1982) 387-411.

Gilmanov A., Sotiropoulos F., Balaras E., A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids, Journal of Computational Physics, 191 (2003) 660-669.

Golberg M.A., Chen C.S., Karur S.R., Improved multiquadric approximation for partial differential equations, Engineering analysis with boundary elements, 18 (1996) 9-17.

Hardy R.L., Multiquadric equations of topography and other irregular surfaces, J. Geophys. Res. 176 (1971) 1905-1915.

Hardy R.L., Theory and applications of the multiquadric—biharmonic method: 20

years of discovery, Comput. Math. Appl. 19 (1990) 163-208.

Hickernell F.J., Hon Y.C., Radial basis function approximation of the surface wind field from scattered data, Appl. Sci. Comput. 4 (1998) 221-247.

Hinatsu M., Ferziger J.H., Numerical computation of unsteady incompressible flow in complex geometry using a composite multigrid technique, Int. J. Numer. Methods Fluids 13 (1991) 971-997.

Hon Y.C., Cheung K.F., Mao X.Z., Kansa E.J., Multiquadric solution for shallow water equations, ASCE J. Hydraulic Engrg., 125 (1999) 524-533.

Hon Y.C., Lu M.W., Xue W.M., Zhu Y.M., Multiquadric method for the numerical solution of a biphasic model, Appl. Math. Comput. 88 (1997) 153-175.

Hon Y.C., Mao X.Z., An efficient numerical scheme for Burgers' equation, Appl. Math. Comput. 95 (1998) 37-50.

Hon Y.C., Mao X.Z., A radial basis function method for solving options pricing models, J. Financial Engrg. 8 (1999) 31-49.

Hon Y.C., Mao X.Z., A multiquadric interpolation method for solving initial value problems, Sci. Comput. 12 (1) (1997) 51-55.

Hon Y.C., Schaback R., On unsymmetric collocation by radial basis functions, Appl. Math. Comput. 119 (2001) 177-186.

Hornung Richard D. and Trangenstein John A., Adaptive mesh refinement and Multilevel Iteration for Flow in Porous Media, Journal of Computational Physics, 136 (1997) 522-545.

Howell Louis H. and Greenough Jeffrey A., Radiation diffusion for multi-fluid

Eulerian hydrodynamics with adaptive mesh refinement, Journal of Computational Physics, 184 (2003) 53-78.

Johnson T.A., Patel V.C., Flow past a sphere up to a Reynolds number of 300, J. Fluid Mech. 378 (1999) 19-70.

Jumarhon B., Amini S., Chen K., The Hermite collocation method using radia basis functions, Engrg. Anal. Bound. Elem. 24 (2000) 607-611.

Kansa E.J., Multiquadrics—A scattered data approximation scheme with application to computational fluid dynamics—I. Surface approximations and partial derivative estimates, Comput. Math. Appl. 19 (1990) 127-145.

Kansa E.J., Multiquadrics—A scattered data approximation scheme with application to computational fluid dynamics—II. Solutions to parabolic, hyperbolic, and elliptic partial differential equations, Comput. Math. Appl. 19 (1990) 147-161.

Kansa E.J., Hon Y.C., Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations, Comput. Math. Appl. 39 (2000) 123-137.

Lancaster P. and Salkauskas K., Curve and Surface Fitting: An Introduction, Academic Press, New York (1986).

Larsson E., Fornberg B., A numerical study of some radial basis function based solution methods for elliptic PDEs, Comput. Math. Appl. 46 (2003) 891-902.

Ling L., Kansa E.J., A least-squares preconditioner for radial basis functions

collocation methods, Adv. Comput. Math. 23 (2005) 31-54.

Liu C., Zheng X., Sung C.H., Preconditioned multigrid methods for unsteady incompressible flows, J. Comput. Phys. 139 (1998) 35-57.

Madych W.R. and Nelson S.A., Multivariate interpolation and conditionally positive definite functions II, Math. Comput. 54 (1990) 211-230.

Micchelli C.A., Interpolation of scattered data: Distance matrices and conditionally positive definite functions, Constr. Approx. 2 (1986) 11-22.

Miller K. and Miller R.N., Moving finite elements, I, SIAM J. Numer. Anal. 18 (1981) 1019-1032.

Moukalled F., Acharya S., Natural convection in the annulus between concentric horizontal circular and square cylinders, J. Thermophys. Heat Transfer 10 (1996) 524-531.

Perng C.Y., Street R.L., A coupled multigrid-domain-splitting technique for simulating incompressible flows in geometrically complex domains, Int. J. Numer. Methods Fluids 13 (1991) 269-286.

Powell M.J.D., The uniform convergence of thin plate spline interpolation in two dimensions, Numerische Mathematik, 68 (1994) 107-128.

Shu C., Richards B.E., Application of generalized differential quadrature to solve two-dimensional incompressible Navier–Stokes equations, Int. J. Numer. Meth. Fluids 15 (1992) 791–798.

Shu C., Chew Y.T., Fourier expansion-based differential quadrature and its application to Helmholtz eigenvalue problems, Commun. Numer. Meth.

Engrg. 13 (1997) 643–653.

Shu C., Differential Quadrature and its Application in Engineering, Springer-Verlag London Limited, 2000.

Shu C., Ding H. and Yeo K.S., Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 192 (2003) 941-954.

Shu C., Ding H., Yeo K.S., Computation of incompressible Navier-Stokes equations by local RBF-based differential quadrature method, CMES, 7 (2005) 195-205.

Shu C., Ding H., Chen H.Q., Wang T.G., An upwind local RBF-DQ method for simulation of inviscid compressible flows, Comput. Methods Appl. Mech. Engrg. 194 (2005) 2001-2017.

Stead S., Estimation of gradients from scattered data. Rocky Mount. J. Math. 14 (1984) 265-279.

Takami H., Keller H.B., Steady two-dimensional viscous flow of an incompressible fluid past a circular cylinder, Phys. Fluids 12 (Suppl. II) (1969) II-51.

Tarwater A.E., A parameter study of Hardy's multiquadric method for scattered data interpolation. UCRL-54670, Sept. (1985).

Tezduyar T.E., Glowinski R. and Liou J., Petrov-Galerkin methods on multiply connected domains for the vorticity-stream function formulation of the

incompressible Navier-Stokes equations, Int. J. Numer. Methods Fluids, 8 (1988) 1269-1290.

Tolstykh A.I. and Shirobokov D.A., On using radial basis functions in a “finite difference mode” with applications to elasticity problems, Computational Mechanics, 33 (2003) 68- 79.

Tuann S.Y., Olson M.D., Numerical studies of the flow around a circular cylinder by a finite element method, Comput. Fluids 6 (1978) 219.

Wang J.G., Liu G.R., On the optimal shape parameters of radial basis functions used for 2-D meshless methods, Comput. Methods Appl. Mech. Engrg. 191 (2002) 2611-2630.

Wright G.B., Fornberg B., Scattered node compact finite difference-type formulas generated from radial basis functions, Journal of Computational Physics, 212 (2006) 99-123.

Wu Z.M., Solving PDE with radial basis function and the error estimation, Advances in Computational Mathematics, Lectures notes on pure and applied mechanics, 202, Guangzhou, 1998.

Zhang X., Song K.Z., Lu M.W., Liu X., Meshless methods based on collocation with radial basis functions. Computational Mechanics, 26 (2000) 333-343.

Zienkiewicz O.C. and Zhu J.Z., The three R’s engineering analysis and error estimation and adaptivity, Comput. Methods Appl. Mech. Engrg. 82 (1990) 95-113.

List of Publications

1. C. Shu, **Y. Y. Shan** and N. Qin, “Development of a local MQ-DQ-based stencil adaptive method and its application to solve incompressible Navier-Stokes equations”, International Journal for Numerical Methods in Fluids, **55** (2007) 367-386.
2. **Y. Y. Shan**, C. Shu and Z. L. Lu, “Application of Local MQ-DQ Method to Solve 3D Incompressible Viscous Flows with Curved Boundary”, CMES-Computer Modeling in Engineering & Sciences, **25** (2008) 99-113.
3. **Y. Y. Shan**, C. Shu and N. Qin, “Multiquadric Finite Difference (MQ-FD) Method and its Application”, Advances in Applied Mathematics and Mechanics, **1** (2009) 615-638.